Research Strategy

Significance

The nested structure of health care delivery and electronic health data

Clinical data scientists are faced with an abundance of useful electronic health data, but limitations of traditional estimators constrain the scientific hypotheses they can explore. Electronic health related data sets are growing rapidly in the number of units of observation and variables observed. This growth implies that we can investigate increasingly fine-grained models. Whereas before we were limited to models of the mean structure and effect, we now want to our models to account for clinical heterogeneity.

The National Anesthesia Clinical Outcomes Registry (NACOR) illustrates the clustered, nested data structure of contemporary health care delivery and electronic health data capture. NACOR is maintained by the Anesthesia Quality Institute and funded by the American Society of Anesthesiologists. We will utilize NACOR as a use case for dissemination in workshops for perioperative data scientists.

Perioperative health care delivery and data capture are nested and clustered. Anesthesia is an illustrative case of the how health care is increasingly electronically documented, which facilitates the continuous collection of physiological data and therapeutic interventions during critical periods. This electronically captured data is joined with surgical and anesthesia procedure codes, International Classification of Disease codes, provider identifiers, patient perioperative risk, outcome and provider compliance assessment. Participating institutions upload this comprehensive file from their anesthesia information management systems directly to NACOR. NACOR contains at present over 30 million individual electronic records of anesthesia care provided and, like similar databases, is growing rapidly. This data set invites health services and outcomes research, but (see letter of support by Dr. Dutton, last director or NACOR and Dr. Kheterpal director of MPOG, the other large perioperative EHR database) their exploitation is limited by the lack of accessible software to investigate important scientific questions with realistic hierarchical models.

Outcomes and care delivered depend on procedure, provider, and patient characteristics. We explain the hierarchical structure of health care delivery with our NACOR model on anesthesia quality[1]. The (dichotomous) outcome (e.g., postoperative nausea) of an anesthetic for a given individual will depend on the surgical procedure the patient is undergoing and under which service, but also on the local institutional culture and indeed the individual anesthesia provider and his or her qualifications and preferences (Figure 1).

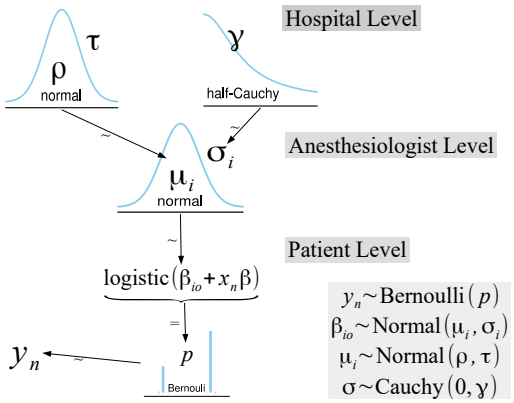

Figure 1: Hierarchical structure of health care delivered in the perioperative setting.

Outcome y_n for the nth patient is a Boolean indicator to be predicted by a logistic regression. We allow the patient-specific intercept β_{oi} to vary according to the ith anesthesiology provider caring for the patient. The provider level mean μ_i and within-provider variance σ_i vary by hospital. x_n is a vector of patient level predictors, and β is a vector of regression coefficients.

Two more examples illustrate the nested structure of health care outcomes: Anesthesiologists may feel more or less inclined or competent to offer regional anesthesia for labor pain. Provision of epidural labor anesthesia varies widely across the nation and within an institution and is predicted by socioeconomic and racial patient characteristics[2;3]. While an average a neurosurgeon may have less blood loss during spine surgery, a particularly gifted orthopedic surgeon may outperform the average neurosurgeon with regards to blood loss.

Hierarchical models capture contemporary health care practice realistically

Hierarchical modeling could transform electronic health records based outcomes research, because the rich spatial and temporal organization of electronic health records is most realistically captured in hierarchical statistical models. However, the estimation of such models is still difficult, both statistically and computationally.

Additional depth of data (simply adding units of observation) will increase the precision of estimates and generally make our clinical data analysis easier. However, there is also more breadth to the data: more subgroups, locations, provider or time granularity than is currently being modeled. Incomplete and noisy measurements cannot easily be incorporated into standard models. Individual patient data do not easily fit traditional meta-analysis[4;5].

Modeling the multifaceted correlations in EHR is reflecting actual clinical practice To realistically model the multifaceted correlations seen in actual clinical practice, we specified a logistic regression model (Figure 1) that predicts anesthesia quality in the NACOR database with regression coefficients that vary by provider, where providers are again nested by service or by hospital[1]. We also need to adjust for the type of surgical procedure, and there are thousands of different surgical procedures performed in the NACOR data. The number of parameters to estimate grows very quickly and so do the potential interactions. This so-called incidental parameters problem violates the assumptions that are necessary for maximum likelihood estimators to be consistent, even if the dataset has a very large number of observations. One solution lies in hierarchical modeling, where we estimate hyper-parameters and allow lower-level parameters to vary across groups[6]. In hierarchical models, we can borrow strength from larger groups and thereby improve estimates for smaller subgroups[7].

Hierarchical models provide efficient inferences with partial pooling Inference based on partial pooling outperforms (a) the no-pooling and (b) the complete-pooling approaches, as can be shown mathematically[8] or via cross-validation[9]: In frequentist terminology, by trading an increase in bias for a reduction in variance, hierarchical modeling can reduce the mean square error both in estimating unknowns and in predicting future outcomes.

(a) The no-pooling approach entails estimate the model separately for each mutually exclusive and exhaustive subsets of the data. Howerver, despite the richness of the EMR data, there are far too many subclassifications (e.g., one model for each type of surgery) to make useful inferences.

(b) Complete pooling constitutes the other extreme of the spectrum, but the equality constraints imposed on the coefficients across all subgroups may be violated by the data-generating process. Ignoring obvious differences in the way the data are generated — such as between patients undergoing tracheotomy versus cesarean section — glosses over the granular detail that is readily available in EHR data.

We choose the middle ground: inference using partial pooling or hierarchical modeling is especially effective for our richly organized NACOR data set because the estimate of each individual parameter is informed by data from all the other patients in our cohort, which improves prediction especially for subgroups with sparse data.[10] Efron explained this apparent paradox well to non-statisticians in the Scientific American[11]. Our co-investigator, Dr. Hall, recently applied this approach to seizure prediction[12].

Borrowing strength is beneficial even in very large data sets In summary, the geographic, spatial and other heterogeneities outlined above can undermine the precision of estimates or predictions even for extremely large clinical data sets. Partial pooling can improve estimation and prediction by borrowing strength from different but related groups[13;14].

Meta-analysis: hierarchical modeling for outcomes research

Systematic reviews and meta-analysis[15] — the synthesis of trial data to support clinical decision making — is another example of hierarchical modeling where traditional software and classical modeling approaches limit progress in data-intensive outcomes research[4]. Evidence synthesis (a more accurate term than meta-analysis) is a powerful tool to pool clinical trials and provide the highest level guidance for clinical care[16;17].

Variance in study design and outcome reporting hamper evidence synthesis

However, studies on perioperative outcomes tend to vary in design and reported outcomes[18], making evidence synthesis challenging with classical frequentist statistical models[19], (see letter of support by Dr. Sacks). Different study designs can make it difficult to perform meta-analysis or meta-regression with classical methods and standard systematic review software[20]. Classical meta-analysis may also underestimate the between-study-variability for small numbers of trials[4;21;22]. Classical meta-analysis is an example of data

with a *single* level of hierarchical grouping: patients' outcomes are grouped within clinical trials[23]. There are however several constraints and limitations with this classical approach to meta-analysis that could be overcome with more sophisticated hierarchical modeling[4;24;25].

Hierarchical models integrate all available data for evidence synthesis

Trial or population characteristics may introduce another level of grouping, for example pooling long-term outcomes after regional anesthesia by surgical intervention[18;25]. Trials may also observe outcomes at several sequential follow-up visits, leading to repeated (correlated) observations grouped by patient that are then grouped by trial, then grouped by surgery in a multi-level meta-analysis.

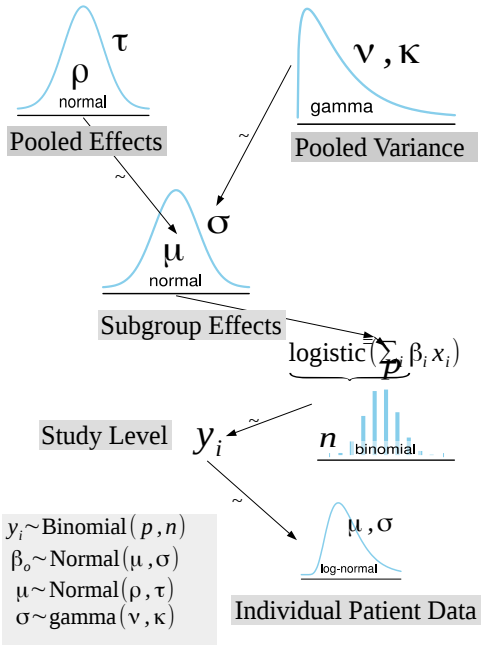

Figure 2: Multi-level meta-analysis.

We build a multilevel hierarchical model to pool individual patient data with continuous and dichotomous aggregate study-level data, clustered at the study level by followup interval and grouped by surgical intervention performed (Figure 2). Besides the three levels of hierarchical modeling just outlined (follow-up observations grouped by patient, patients grouped withing trials, trials within population), Additional hierarchical structure may be needed reconcile the substantial between-study variance in findings[4]. Evidence synthesis may seek to integrate continuous and dichotomous outcome measures[26], since meta-analysis requires *all* available data sources[20]. However, short of having access to all individual patient data, the integration of dichotomous outcomes and continuous outcomes is challenging with traditional software[4;27].

The antithesis of complete versus no-pooling limits meta–analysis of perioperative outcomes To illuminate how the false dichotomy between complete versus non-pooling also limits evidence synthesis, we consider meta-analysis of trials with repeated observations on the outcome of interest. The follow-up intervals (at which perioperative outcomes are reported) often vary in different studies. Additionall, some studies report repeated measures, while others only report a single terminal observation, which leads to the same issue of (a) complete pooling versus (b) no-pooling in evidence synthesis[27].

(a) Complete pooling: Evidence synthesis of all effect estimates at all time points irrespective of follow-up time fails to consider the correlation of repeated measure. Obviously, it is usually inappropriate to ignore the fact that the effect estimates may depend on how long after the intervention the outcome was observed.

(b) No-pooling: Conducting separate meta-analysis for each follow-up visit would undermining the main strength of meta-analysis, in addition to the estimates being statistically unjustified and imprecise.

Clustering can bias estimation of confidence intervals. To correctly estimate confidence intervals even in a simple Student t-test, we have to consider if the data are independent, which they would not be if they are observed for the same patient multiple times. Failure to take into account dependence in clustered observation may lead to incorrect inferences. Correction to confidence intervals, such as weighted jackknife methods, can improve matters, but such approaches are often ignored in EHR research and they depend on characteristics that may not hold in faceted EHR data

Clustering and correlations can bias clinical inferences in meta-analysis A serious bias results from failing to consider correlations between outcomes in the interpretation of meta-analyses. Even Cochrane Reviews, in clinical practice are often reduced to "shows a significant effect" vs. "shows no significant effect". But modeling rather than ignoring the correlations between outcomes could turn a significant effect into a non-significant effect (or vice versa) and the origianl model could overestimate or underestimate the variability of correlated outcomes. For example, assume the estimated overall relative risk (RR) is 0.8 for an intervention across all studies pooled. The estimation of a 95% confidence interval of $0.55 - 1.15$ would lead to the inference of "no statistically significant effect" and thwart further attempt to study this intervention, while a 95%

confidence interval of $0.7 - 0.9$ for the effect estimate may lead to widespread adoption of this therapeutic approach, with huge clinical impact.

Ecological, disease and geographic study level characteristics can influence inferences There are other forms of ecological bias to be considered in meta-analysis besides clustering by reported time endpoint. Geographical or historical settings, similar surgical procedures, or related diseases will lead to correlated outcomes in patient cohorts[25;18;4;27]. Thus, the same false dichotomy of complete pooling versus no-pooling limits meta-analysis for perioperative outcomes and hinders evidence-based medicine. The locations of the estimates, their precision, and the inter- and within-study variability may all differ considerably contingent on disease, procedure, or other study level characteristics[18;4;27].

Partial pooling across similar populations The principle hierarchical modeling for evidence synthesis is analogous to outcome research in general where we consider effects that more prominent in one trial or subgroup of the population and not as strong in another similar, but distinct, trial or subgroup. Examples include the meta-analyses of a treatment effect in conditions with different but related etiologies and the spectrum of HIV-related, idiopathic, diabetic, and traumatic chronic painful neuropathy[4]. Information in one subset or study population can and should inform estimates in other similar populations to some estimated extent in a hierarchical meta-analysis. The principle applies in other fields of medicine, e.g., critical care[27], and becomes more relevant if the average effect sizes, the precision of effect estimates, and the inter- and within-subgroup variability differ considerably from the subgroups used to obtain the aforementioned findings.

Hierarchical modeling to improve data-intensive outcomes research Hierarchical models are thus a useful tool for analyses of heterogeneous perioperative outcome data. This is true for meta-analysis of long-term studies with varied design to better inform clinical decisions[26;28] and for outcomes research in our hierarchically structured contemporary health care delivery system. Advanced hierarchical models can be difficult to fit with traditional software but should be made more accessible (see letter of support by Drs. OMalley, Sacks and DiMaggio).

Integrating incomplete data into clustered EHR modeling

Too often data scientists either (1) fit sophisticated models, but limit their analysis to complete cases, ignoring any missing or incomplete data or (2) impute the missing data using overly simplified models of the scientific question of interest (see letter of support by Dr. Mirhaji). Hierarchical modeling can synthesize the model(s) for the missingness with the model for the outcome(s).

EHR data are not missing at random. Physicians chose which test to administer in order to inform specific therapeutic decisions. Different types of data are recorded in different clinical settings (e.g., arterial lines may not be permissible on the floor, vitals are recorded in greater detail and more frequently in high dependency units like the ICU). In the Innovation section, we explain how both missing data imputation and clustering can be integrated into a single model that incorporates auxiliary data.

Innovation

From about 1990 to about 2010, Markov Chain Monte Carlo (MCMC) was the most important breakthrough for applied statistics in all quantitative subfields, including medical research. Hamiltonian MCMC has become prominent in the past five years and is likely to drive the next second generation of applied statistcs that utilizes MCMC. Compared to first generation MCMC such as Gibbs samplers and random-walk Metropolis-Hastings, Hamiltonian MCMC converges to the posterior distribution in fewer iterations, exhibits negligible dependence between consecutive draws from the posterior distributions, and outputs auxiliary information that indicates when something has gone wrong. These innovations allow applied statisticians to fit more sophisticated models, such as hierarchical models, and to expect results that are statistically reliable.

The *Stan* project is at the forefront of the Hamiltonian MCMC revolution. *Stan* provided the first general implementation of Hamiltonian MCMC in 2012 and incorporated two critical features that overcame the main inhibitors to its widespread use. First, *Stan* is almost self-tuning, so researchers can focus on specifying their models rather than wasting time trying to tweak algorithms they have a very limited understanding of in

hopes of obtaining acceptable performance when estimating their models. Second, *Stan* utilizes automatic differentiation, which relieves researchers of the burden of supplying a function that calculates the gradient of the posterior distribution with respect to the (perhaps thousands of) parameters in their model.

Despite all these technological advances, *Stan* is currently not used widely enough in medical research, although it is now well-known in other subfields. Our grant proposal is innovative in the sense that it will break down three barriers to using *Stan* productively in data-intensive medical research.

First, up until the last few months, the only way to use *Stan* was to write the model from scratch in the Stan language, which requires considerable expertise in probability theory and substantial experience using *Stan* in order to estimate a model as complex as a hierarchical regression with many grouping factors. Today, it is quite possible to use our first stable version of our new *rstanarm* package for *R* to estimate pre-written pre-compiled hierarchical models in the Stan language that are very general, highly optimized, and have been extensively tested by us. Thus far, *rstanarm* may be too narrowly focused on applications in the social sciences and more work is needed to enhance *rstanarm* for medical research by incorporating suggestions from medical researchers, (see letter of support by Dr. DiMaggio).

Second, although *Stan* can treat missing values on continuous variables as additional unknowns of the posterior distribution, it is not particularly easy to do so in the Stan language. Moreover, dealing with missing values on discrete variables is downright tedious. By the end of the grant, we would like models provided by *rstanarm* to handle missing data seamlessly but in a statistically appropriate fashion. As an intermediate step, we will innovate by overhauling our *mi* package for multiple imputation of missing data to utilize the advanced model-fitting functions provided by the *rstanarm* package. Doing so will allow missing values to be imputed in such a way that it preserves the rich grouping structure in EMR data, which is currently ignored by *mi* and all other algorithms for multiple imputation.

Third, a variety of things can nevertheless go wrong, and it is critical that such problems be recognized and overcome. For example, *Stan*'s self-tuning is a work in progress and in any particular research project may require the user to tweak one setting. Or one reparameterization of the model may be much more computationally efficient than another logically equivalent parameterization. Finally, the model may simply fail to fit the data well. We will innovate by continuing to develop *shinystan*, which is a web-application that integrates with *rstanarm* and provides a plethora of tools to visualize the parameter estimates, diagnose the supplemental Hamiltonian MCMC output, and make scientific inferences. Without *shinystan*, understanding the output of an advanced hierarchical model may be rather difficult for all but the most computationally sophisticated medical researchers.

Approach

We propose to further develop three accessible software packages (*rstanarm shinystan*, and *mi*) for the programming language and software environment *R* so that they are reliable, and trustworthy being extensively tested in different setting and applications. Each of these *R* packages are currently available for free to any researchers via the Comprehensive R Archive Network (CRAN) software repository. Our software development project will be guided by a multidisciplinary team, which will conduct its work through regularly scheduled weekly meetings and collaborate online via Github[42], Google Hangouts, and email.

These software products will facilitate hierarchical modeling for data-intensive outcomes research — even if some of the data are incomplete — and make it easy to visualize, diagnose, and understand the model output. We will also promote and disseminate these innovations through presentations, graduate-level teaching, workshops, online tutorials, books, YouTube videos, and other publications.

The state of software for hierarchical modeling

The Bayesian inference Using Gibbs Sampling (BUGS) project got its start in 1989 and has since grown into a family of related software such as WinBUGS (for Windows), Just Another Gibbs Sampler (JAGS), and OpenBUGS. Bradley Carlin, a prominent Bayesian professor of biostatistics, stated

> MCMC freed Bayes from the shackles of conjugate priors and the curse of dimensionality; BUGS then brought MCMC-Bayes to the masses, yielding an astonishing explosion in the number, qual-

ity, and complexity of Bayesian inference over a vast array of application areas, from finance to medicine to data mining.

Although there are many examples of BUGS models available on the internet, a BUGS model essentially has to be written from scratch for each new research project, which is too much to ask of many medical researchers. At the same time, it is widely known that it is all too easy to write a model in the BUGS *language* that is unfeasible for the BUGS MCMC *engines* to sample from efficiently. In particular, BUGS models with many parameters are difficult to fit because BUGS typically can only update the state of the Markov chain one parameter at a time.

Consequently, in recent years many hierarchical modelers have shifted back to essentially frequentist estimators that choose the common parameters to maximize a (perhaps penalized) likelihood function that integrates out all the group-specific unknowns. Examples of this approach include the *lme4* package for R, the PROC MIXED subroutine for SAS, and the `xtmixed` and `gllamm` functions for Stata. Researchers can often not fit the models they find scientifically most credible, (see letter by Drs. Kheterpal and Dutton and letter by Dr. Sacks).

The aforementioned approaches have their pros and cons. One on hand, they may make multilevel modeling accessible to a larger number of researchers because the user-facing functions employ familiar simple notation, simplifying model specification. A user need only specify the outcome and the predictors, the grouping structure, and perhaps some options in order to estimate a model that has been pre-written in a compiled language. Thus, users need not write customized modeling code like they would with BUGS.

On the other hand, these frequentist approaches do not provide measures of uncertainty for the group-specific unknowns, such as intercepts and coefficients. Indeed these group-specific unknowns are not considered parameters to be estimated by rather random variables to be predicted. In medical treatment centers, doctors may well be interested in the implications of a researcher's model for a particular patient, but the uncertainty in the patient-specific unknowns cannot be easily quantified with these frequentist estimators. In addition, the estimated standard errors for the common parameters are obtained under the assumption that the group-specific random variables are known. Consequently, when there are many group-specific unknowns, the true uncertainty in the estimates of the common parameters may be understated. Finally, complicated hierarchical models often encounter computational problems that prevent the optimizer from finding the maximum in the interior of the parameter space.

The *Stan* language, mathematical library, and algorithms pick up where BUGS left off. An enormous variety of hierarchical models can be specified in the Stan language, but the researcher needs to possess considerable skill in computational statistics in order to specify such an advanced hierarchical model. Given the hierarchical specification, the Hamiltonian MCMC engine included in *Stan* can sample from the posterior distribution efficiently, even if there are hundreds of thousands of parameters to estimate. With MCMC estimators, both the common and the group-specific parameters are part of the joint posterior distribution, so their uncertainty can easily be summarized by the standard deviation of the posterior samples, credible intervals, and so forth.

What is needed now, what we propose to develop further, what we implemented in the first functional version of our software packages *rstanarm*, is to tack back in the direction of the *lme4* package in order to make the power of *Stan*'s Hamiltonian MCMC engine available to a broader subset of medical researchers who are interested in estimating hierarchical models (see letter of support by Drs. Sacks and O'Malley). This step has been partially accomplished by the first public release of the *rstanarm* R package on CRAN in mid-January of 2016. The `stan_glmer` function in the *rstanarm* package adopts the same syntax for specifying hierarchical models as the `glmer` function in the *lme4* package, which is a slight generalization of the now standard R syntax for specifying models via a formula that refers to the outcome and predictor variables that are columns within a data.frame. The transparency of *rstanarm* will make medical researcher more reproducible and reliable enough even for federal regulatory processes.

For example, to specify a linear model where the intercept and the coefficient on the `x2` variable are allowed to vary by group, a user would specify something like `y ~x1 + (1 + x2|g)` where the dependent response

variable y is on the left of a tilde ($\sim$) operator; the independent terms, $x_1, x_2...$ are on the right and separated by the $+$ operator. Group-specific terms are distinguished by vertical bars $|$ separating expressions for design matrices from grouping factors.

Thus, the researcher need not express a hierarchical model in the Stan language in order to estimate that model with *Stan*'s Hamiltonian MCMC engine because *rstanarm* comes with abstract models have already been written in the Stan language by our team and, of equal importance, have been tested to verify that the models work correctly. In short, *rstanarm* combines the user-friendliness of *lme4* with the computational efficiency of *Stan* and is integrated with the other parts of the Stan ecosystem — all of which are open-source and free for anyone to use — such as the *shinystan* package for visualizing and diagnosing problems in the estimates.

Priors *can* incorporate subjective beliefs or existing information about a parameter, which we discussed in more detail under significance[6][7]. Bayesian models require priors and *rstanarm* by default adds independent weakly informative priors on the coefficients of generalized linear models, but more informative priors can be specified by the user. Especially in hierarchical modeling, regularization with priors can help convergence and improve estimates and predictions[9]. In *rstanarm*, the user can choose from a broad array of distribution. For example, the intercept one might choose the Student t distribution, which approaches the normal distribution as the degrees of freedom parameter approaches infinity and approaches the Cauchy distribution as the degrees of freedom parameter approaches one. Moreover, the degrees of freedom can be estimated in hierarchical models, which allows the data to determine what tail heaviness is appropriate.

To date, *rstanarm* has only had one official release, so there are some features that are incompletely implemented and a trickle of small bugs have already been discovered by users, which may continue for some time as *rstanarm* gets more exposure in different research contexts. Nevertheless, the foundation of *rstanarm* appears to be solid and it is ready to be built upon in order to make even more sophisticated and flexible hierarchical models accessible to medical researchers. Dr. Goodrich and Mr. Gabry stand ready to collaborate with Drs. Andreae, Hall, and Gong, as well as the broader medical research community to implement more specific models and to tune them for data-driven outcomes research. In response to one of Dr. Hall's first suggestions, we will develop support for change-point models in *rstanarm*.

The state of software for multiple imputation of missing data

Multiple imputation is now an old idea that under certain assumptions about the missingness mechanism(s) that serves as a first step toward statistically justified inferences when there are missing data. First, the researcher uses a model or sequence of models to impute values for each observation where the data were originally missing. This process is repeated multiple times in order to obtain several completed data sets. Second, the researcher analyzes all of the completed data sets as if the data were fully observed from the outset and applies some rules to combine the estimates from the several completed datasets.

Multiple imputation is not ubiquitous in applied research, although it is now fairly commonplace in large part due to the availability of functions in Stata and in *R* packages that try to automate this two-step process as much as possible. However, the open secret of multiple imputation software is that the models used to impute values in step one are very simplistic compared to the models that are often fit to the completed data in step two. In other words, the imputation models essentially assume the data are a simple random (albeit incomplete) sample from a population, while the substantive models often try to exploit the obvious grouping structure that exists in many data-generating processes, such as EMR. As a result, the imputation process tends to dilute the group structure that is essential to the analysis.

There are two major imputation algorithms. One assumes that all of the variables are jointly multivariate normal and estimates its mean vector and variance-co-variance matrix via an iterative process that draws values from a conditional multivariate normal distribution that conditions on all the observed data for each observation. This approach is simple, but is not theoretically appropriate for binary or categorical data, which cannot be generated as part of a multivariate normal. Nevertheless, many researchers use this algorithm even when their variables are largely categorical on the grounds that doing so is better than not imputing the missing values at all.

The second major imputation algorithm specifies a sequence of conditional distributions where each model in the sequence pertains to the conditional distribution of one variable with missingness given (some subset of) the other variables. When each model is estimated, imputed values are drawn from a univariate conditional distribution that can be tailored to each variable depending on how it is measured. Thus, the second approach avoids the embarrassment of imputing continuous values for binary or categorical variables but suffers the embarrassment that the sequence of conditional distributions generally do not imply any valid joint distribution over all the variables with missingness. Also, if there are many variables with missingness it is very tedious for users to specify a conditional model for each incomplete variable.

Our *mi* package, which we propose to further develop, takes the latter approach but uses simplistic imputation models that ignore the grouping structure that user may know exists in the data-generating process. As an initial step, we would like to reformulate the *mi* package so that is uses the models in the *rstanarm* package to impute. Hamiltonian MCMC is generally too slow for this task when there are many variables with missingness, but *Stan* also includes some much faster algorithms that are intended to provide draws from a reasonable approximation to the posterior distribution. When this initial step is completed, *mi* will be able to quickly impute missing values from a model that takes into account, rather than ignores, the same grouping structure that the researcher plans to exploit when the substantive model is estimated on the completed data sets.

As the project progresses, we would like to include models in the *rstanarm* package that simultaneously model the missing values, the missingness mechanism, and the outcome of interest, which typically would include some specification of the grouping structure. This approach implies a well-defined joint posterior distribution of the model parameters and the unknown data and thus should yield estimates and predictions that are not only statistically justified but also more precise.

Heterogeneous and incomplete clinical data may limit prediction and implementation.

Variables with strong predictive power may not be recorded for all patients or may be missing for the time window needed for prediction, which is a critical limitation in the development of prediction algorithms and implementation of the therapeutic interventions (see letter of support by Dr. Mirhaji). Yet, incomplete data are the hallmark of EMRs. Hierarchical models for incomplete data give valid estimates *if and only if* the missingness mechanism is ignorabe, which is to say that the parameters for the missing data mechanism are independent from the parameters in the main model for the outcome, and the data are either missing at random (MAR) or Missing Completely At Random (MCAR)[31]. Indeed, these assumptions are not reasonable for EMRs. In our example, only significant respiratory co-morbidity and symptoms will prompt physicians to request arterial blood gases (ABG). Trying to impute missing ABG data using traditional multiple imputation algorithms would hence lead to biased imputations. Imputation using auxiliary data can help overcome this limitation, as outlined below.

Auxiliary data can be used to impute incomplete medical records.

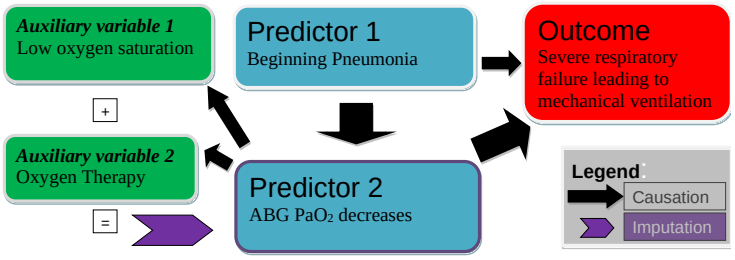

Figure 3: Auxilliary data to impute missing information

Auxiliary data are additional information available in the form of variables known to be correlated with the missing data of interest[32;33]. Figure 3 illustrates how we can impute incomplete data from auxiliary information[34;35]. We know pneumonia impairs oxygenation, by causing respiratory failure, for example. If arterial PaO_2 (oxygen tension) is missing because arterial blood gases (ABG) have not been obtained, we can impute the incomplete data from oxygen therapy and/or peripheral oxygen saturation[32]. This approach avoids the perils associated with missing at random (MAR) assumptions, when fitting a non-ignorable missingness model[36]. Adding auxiliary variables not included in the main model for multiple imputation, in other words using additional information that is correlated with the missing outcome is an emerging approach to help correct bias[37;38;39], often relying on Bayesian methods[40;41]. Joint hierarchical modeling, including auxiliary data to impute incomplete patient records, will

improve the prediction model and facilitate the implementation of the prediction algorithm[32] for our use case to predict inpatient respiratory failure from EHR in Montefiore Medical Center using the data from Dr. Gong's pragmatic trials APPROVE and PROVECheck.

The state of software for visualizing MCMC output

Almost all MCMC software has some capability to make graphs from MCMC output. But none are as user-friendly or as capable as *shinystan* for exploring and troubleshooting the output of hierarchical models.

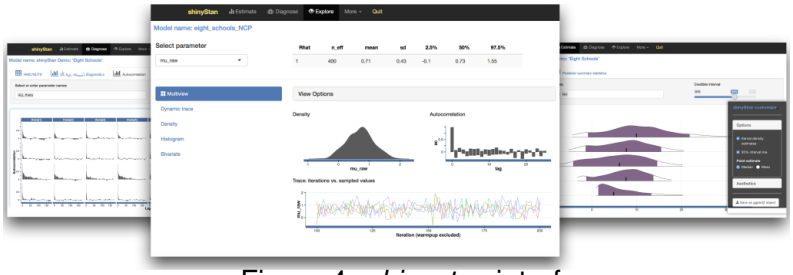

Figure 4: *shinystan* interface

The interactive user interface of *shinystan* shown in Figure 4 demonstrates how easy it is to do posterior predictive checks of the model against the observed data, which are routine if the model was estimated by any of the functions in the *rstanarm* package. If a hierarchical model fails to converge, ascertaining why is crucial to troubleshooting. *shinystan* is already unmatched in combining ease of interactive graphical exploration with sophisticated graphical rendering to explore correlation between parameters, auto-correlation in the draws, the number of steps that the Hamiltonian MCMC sampler took on each iteration and many other convergence diagnostics.

Although *shinystan* is already available on CRAN and is a fully functional package, it will need to undergo considerable improvements of the interface and under the hood during this project. Posterior predictive checking is one area where the need for additional options and functionality. We also want to expand the existing collaborative model sharing via the internet. We will not only further develop *shinystan* to make it more robust and scale it to work faster for larger models and data sets. We will continue to develop novel graphical methods to assess model convergence with a special focus on hierarchical modeling. Finally, a major goal for *shinystan* towards the completion of our project is to be able to save the *R* code that generates each of these graphics. When that capability is added, users will be able to include the plot-generating code in their *R* scrips so that the graphics can easily be updated whenever new observations are added to a data set or when the model is tweaked. This supports transparent reproducible research much better.

As we further develop the interface we will place special emphasis on the concept of affordance in our "intuitive" design of the *shinystan* graphical user interfaces, e.g., in employing strong visual clues, provide clickable buttons and tabs, sliders, and other hands on interactive controls[72]. Active exploration by *shinystan* users will reveal nested and sequential affordances, for example, print options appear only in context, say when the cursor moves over the output button of a graphic, avoiding visual clutter on the screen with irrelevant information[73]. We will rely on user feedback to optimize the practical utility of *shinystan's* implementation and user interface.

Project scope and goals

In summary, we propose to further develop, test, and harden our three software packages: *rstanarm*, *shinystan*, and *mi*.

rstanarm for accessible hierarchical modeling

The first proposed software package *rstanarm* allows data scientists to specify the most common applied hierarchical regression models and estimates them via *Stan*'s implementation of Hamiltonian MCMC algorithm. Researchers do not need to understand the underlying intricacies (e.g., auxiliary parameterizations) that have been implemented in order to accelerate convergence and do not need to build complicated data structures to keep track of group indexing. *rstanarm*'s syntax makes hierarchical model building easy because it is the same as the familiar notation for specifying models in *R* and popular *R* packages such as *lme4*[63]. We will add additional models to *rstanarm* so that it will become the premiere tool for hierarchical modeling in medical research.

shinystan for interactive graphical exploration and diagnosis of MCMC output

The second *R* package, *shinystan*, is a graphical user interface for interactively exploring any model estimated by MCMC. *shinystan* provides multidimensional visual tools for any analyzing high-dimensional MCMC output, particularly for *rstanarm*. *shinystan*'s web-application interface is user-friendly and requires minimal training for novices.

mi for multiple imputation of missing data

The third *R* package, *mi*, performs multiple imputation of missing data by specifying a simplistic model for each incomplete variable. We propose instead using the models provided by *rstanarm* to provided the basis for the imputations, which would allow us to specify the group structure that is evident in EMR data. Ultimately, we propose enhancing *rstanarm* so that it can estimate models in a statistically principled way, even when some of the data are incomplete.

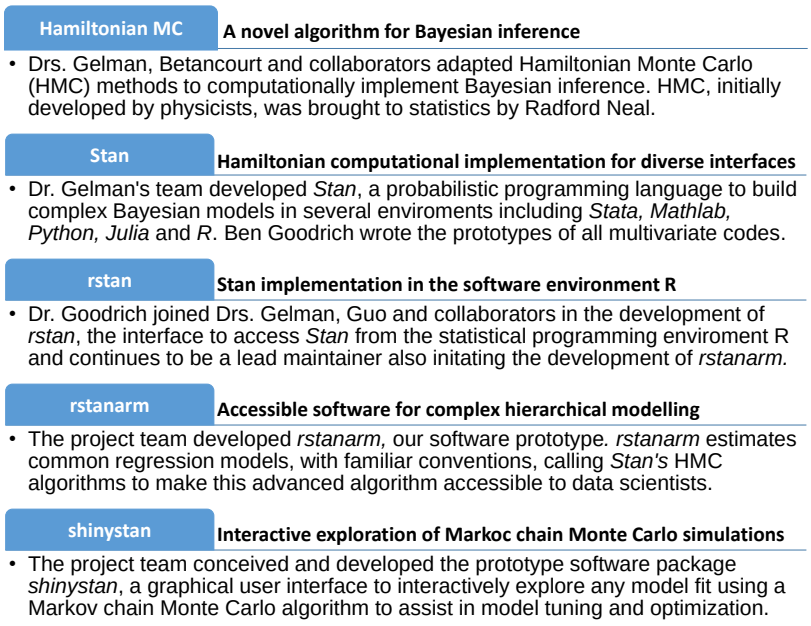

Figure 5: Software development trajectory

The trajectory leading to the development of *rstanarm* and *shinystan* The National Institute of Health, the Center for Disease Control and the National Science Foundation, (NIH: 5R01GM074806, 5KL2TR001071, UH2-HL125119, CDC: U01 OH010711-01 and NSF: SES-1205516) were among the many federal institutions funding Stan[62] development and its empirical applications. The work proposed here is thus a direct continuation of the aforementioned work that developed and implementing ground-breaking algorithms that are capable of estimating complex hierarchical models. Figure 5 to the above left outlines the trajectory that led to the current project proposal.

Prior work in statistical software development and data-intensive outcomes research Dr. Andreae published several meta-analyses and synthesized the evidence from clinical trials by pooling aggregate and individual patient data, when published results were insufficient for classical meta-analysis[26;18;4;43;44]. Dr. Andreae, Hall, and collaborators used the software *Stan*, *rstanarm* and *shinystan* to build a multilevel hierarchical model to investigate health care disparities and quality of anesthesia delivery in the large National Clinical Outcomes Registry maintained by the American Society of Anesthesiology[1].

Dr. Hall has also published on missing data imputation[12;36;36] and is internationally recognized for the development and application of change point models in epidemiology and surveillance[47;48;49;50;51]. More recently, Dr. Hall has played a major role as the lead statistician for the World Trade Center (WTC) Health Program at the Fire Department of the City of New York, supervising data analyses based on medical records[52;51;53].

Dr. Goodrich build the current version of the R package *mi* as a postdoctoral researcher working with Dr. Gelman[45] who is internationally recognized as a leader in Bayesian statistics, hierarchical modeling, and data imputation[59;29;60;9]. Dr. Gelman oversees *Stan*[46] project, which which provides the foundation for the proposed work and Dr. Goodrich has been one of several core developers supplying that foundation since 2011. Dr. Betancourt is a consultant on this grant with his special expertise on the differential geometry of the underlying Hamiltonian MCMC algorithm in *Stan* that allows for automated tuning heuristics[61].

Dr. Goodrich is now the maintainer of the *rstan* R package and the *rstanarm* package that extends it, which was written entirely by Dr. Goodrich and Mr. Gabry. Mr. Gabry is the maintainer and was essentially the principal code writer of the *shinystan* R package.

Dr. Gong and Gelman's have an promising collaboration involving the ongoing NIH funded trial to predict

and improve respiratory outcomes after intubation based on real time electronic medical records. This trial is just one of many examples of Dr. Gong's leading role in applied data driven outcomes research[54;55;56;57;58].

Collaborative software development and computer resources

We will follow accepted best practices, proven methods, and industry standards for open-source software design, construction and implementation. For example, when building *Stan* and its ecosystem of interfaces, we have trie to follow the rules of clarity, composition, separation, simplicity, transparency, etc. set form by Raymond[75].

The source code of *rstanarm* and *shinystan* are managed through the Git version control system [42] via the online software repository Github for integrated issue tracking, progress milestones, and code history tracking[76]. We will use the Git process[77], as we did sucessfully for the development of *Stan*[62]. This process is based on collaborative code review, where new features or functions are developed creating a branch; proposed changes and additions are discussed following a pull request on Github. The ability to write to the repository is restricted to the core team, while any user can propose patches to address software bugs or suggest improvements. Further commits are commonly added after substantial feedback and testing before merging the branch. On Github, altered syntax is highlighted and changes are transparent and revertible, making collaborative software development easier. Github's distributed structure implies that everybody cloning the repository generates a backup. We prototype and test additional functions and features before inserting them in any package update.

The development, extension, testing and maintenance of our packages is a collaborative effort by our multidisciplinary team with communication and input taking place in several venues. Public user groups and discussion are archived and freely accessible. Confidential deliberations, (e.g., regarding grant related matters), are held in private restricted groups. We will hold weekly video conferences to prioritize issues and discuss progress.

Montefiore Medical Center and Columbia University will provide the computer cluster access required for the computationally intensive models and provide server space to house webpages. Clinical Research Informatics at Montefiore will provide scale-able remote access via a secure virtual private network to up to 32 Xenon Intel processors in parallel on a windows virtual machine with up to 128 GB RAM to meet the variable needs of the project throughout the course of the grant.

As also discussed under the subsection on human subject protection and in the letter of support by Dr. Mirhaji, some of the data sets to be used for this project cannot be completely de-identified. Rather, they are 'limited data sets' that still contain some patient identifiers (e.g., age in years or zip code). Hence, the computer clusters housing the data and any computers that we use to analyze the data have to be compliant with the Health Insurance Portability and Accountability Act (HIPPA. Clinical Research Informatics at Montefiore will house any data set falling under HIPPA and guarantee compliance with relevant federal data safety and privacy regulations (see letter by Dr. Mirhaji).

Dissemination and unimpeded utilization of the products of this project

The goal of this project is to encourage widespread adoption of cutting-edge hierarchical modeling for data-intensive outcomes research by clinical scientists and to support the use of our open source applications. The software developed for this grant will hence all be incorporated into the public repository CRAN, of the open-source R Project for Statistical Computing. Our source code and documentation will be distributed under the least restrictive open-source licensing terms possible: R's licensing is *copyleft* under the GNU General Public License. *Copyleft* refers to licensing arrangement that allow for the software can be used, modified and/or distributed freely on condition that anything derived is bound by the same condition. The benefits are that our software (1) will be freely available to researchers and the general public, including for commercial use, (2) shall remain freely available to the public and may be freely extended, (3) may be incorporated into the other tools that also are licensed under the same terms, (4) can be maintained if the original developers are no longer able to, and (5) can be enhanced based on user-provided feedback for bug-fixes, examples, and enhancements.

Prior team experience and exposure in statistical and quantitative teaching and dissemination

Our project team have ample experience in teaching, publishing, and promoting software. The Co-Principal Investigators teach in graduate programs focused on teaching quantitative skills to social science and biomedical audiences. Our workshops on statistical and research methods are sought after, and we have been invited to national and international statistical and biomedical meetings. Our team presented workshops at a wide spectrum of scientific and statistical meetings ranging from clinically oriented conferences like the Annual meeting of the American Society of Anesthesiologists to machine learning venues to the Annual Conference on Neural Information Processing Systems (NIPS). Dr. Gelman maintains a very popular blog. Presentations and products by our team have been featured on YouTube and blogs by others.

Books, tutorials, workshops lectures and outreach to disseminate hierarchical modeling

Throughout the duration of the project we will disseminate the software packages in tutorials, workshops, lectures, books and other forms of outreach. In particular, we will offer hands-on workshops at national and international for small groups based on our applied use cases as outlined also in the budget justification. These will be interlinked and supported by online material, for example YouTube videos promoting or explaining and motivating the algorithms underlying *Stan*. We will make detailed technical and practical tutorials and vignettes available online, interspersing code, graphics and detailed step-by-step explanations using *R's* R Markdown Dynamic Documents for R, blogs, for example our blog on Statistical Modeling, Causal Inference, and Social Science. In addition, *rstanarm* and *shinystan* will be featured in the next edition of Bayesian Data Analysis with many examples[9].

Mechanisms for incorporating feedback and user reported corrections into the software.

We already have very active user groups around *Stan* and with frequent online and face-to-face meetings. We will incorporate more of *rstanarm* and *shinystan* in the future to discuss implementation issues, engage advanced users in the development of our packages and/or in designing courses around our models and software, but in particular to solicit their feedback. The user forums have served also to assist novices in implementing their models in our software. Like our existing Wiki on Github for *Stan*, we will take full advantage of the functionality of GitHub for engaging users in collaborative software development[76], including further development of the *rstanarm* and *shinystan* Wikis and discussion of issues with users.

Timeline and potential problems

Project Year	rstanarm	innovation	shinystan
First	organize project team, update rstanarm package on CRAN, establish online user group	Improve online demo version of shinystan, more flexible posterior predictive checking tools	
Second	add levels to rstanarm generalized linear model, implement basic meta-analysis functions	develop new visual tools for graphical exploration of model convergence and model fit	
Third	flexible multilevel meta-analysis functions, integrate rstanarm algorithms in multiple imputation package	Develop visualization for incomplete data, Integrate some shinystan functions in Stan	
Fourth	develop and implement basic missing data integration into rstanarm, start implementing advanced multilevel change point models	Interactive visualization tools to test model fit for advanced multilevel models	
Fifth	scale missing data integration to larger datasets	Incorporate shinystan functions into the main software Stan	

The Timeline delineates the planned project progress by year. We begin by setting up the project team in the first year, establishing our online user interface and updating the *rstanarm* package in CRAN. In year two and three, we will sequentially add additional models and provide advanced tools for visualization. We start to develop additional functions for meta-analysis and change-point models, scaling eventually these also to multilevel models. In the fourth project year, we will begin to integrate missing data algorithms into *rstanarm* and finally scale the packages to large datasets in year five.

Some colleagues are reluctant to make powerful statistical tools accessible to the novice users who may not fully understand them. We counter however that the simplicity of the *rstanarm* syntax will allow novice users to learn to specify more realistic models by making the underlying hierarchical model structure more transparent. *rstanarm* will also allow outcomes research to be more transparent and reproducible. The integration of *mi* with *Stan* through the use of *rstanarm's* pre-compiled will greatly improve incorporation of incomplete and missing data into advanced coherent modeling. Finally, *shinystan* will contribute to the ease with which advanced hierarchical models and MCMC results can be shared, questioned, and discussed.