Biographical Sketch

NAME: Benjamin Goodrich

eRA COMMONS USER NAME: B_GOODRICH

POSITION TITLE: Lecturer in Discipline in Political Science at Columbia University

EDUCATION/TRAINING

INSTITUTION AND LOCATION	DEGREE	COMPLETION DATE	FIELD OF STUDY
Emory University, Atlanta, GA	B.A.	05/2001	Political Science
Emory University, Atlanta, GA	M.A.	05/2001	Political Science
Harvard University, Cambridga, MA	Ph.D.	05/2010	Government and Social Policy
Columbia University, New York, NY	Postdoctoral	06/2013	Applied Statistics

A. Personal Statement

Although my degrees have been in political science, I am essentially a computational statistician with interests in multiple applied fields. I will be a co-principal investigator on this grant proposing to build accessible software to facilitate the building of complex hierarchical models for data-driven clinical outcomes research.

My co-principal investigator, Michael Andreae is a perioperative physician; I feel he and his colleagues at the Albert Einstein College of Medicine complement me very well with their extensive experience in clinical medicine and applied statistical modeling for outcomes research. Hence my focus in this project will be on software development (*rstanarm* and *shinystan*) and advanced hierarchical modeling for our use cases, as well as teaching and dissemination to the high end users through workshops and online user groups. But we will lead the project jointly and in a close nit collaboration including frequent interaction with the senior co-investigators, for which I am well prepared also thanks to my prior experience as a co-investigator in an NIH funded project.

I have been a core developer of the Bayesian statistical software project called *Stan* almost since its inception in 2011 and am the maintainer of two packages for the statistical software environment R in which we implement our new software *rstanarm* and *shinystan*. (1) *Rstan* provides the interface to the Stan library in the R environment. (2) *mi* is a software package for multiple imputation of missing data I developed after completing my Ph.D., as postdoctoral researcher on an Institute for Education Sciences grant.

This grant application proposes substantial enhancements to our aforementioned existing software projects (*Stan*, *Rstan* and *mi*), which have been developed in collaboration with Andrew Gelman and other researchers at and outside Columbia University. *Stan's* selling point is the Hamiltonian Monte Carlo algorithm that makes its effective sampling rate orders of magnitude faster than any other existing software. *Stan* and *Rstan* have been well-received in a variety of scientific fields. In the past five years, *Stan* has already become prominent in applied statistics

and the social sciences, and we would like to bring these achievements to the field of medicine and expand upon them during the next five years.

As a lecturer at Columbia University, I teach mostly advanced hierarchical modeling and data imputation in their Quantitative Methods in the Social Sciences graduate program. Between semesters, I give workshops to promote and disseminate *Stan*, for example, I taught a two-day workshop on *Stan* at the headquarters of The Climate Corporation, which employs dozens of data scientists who study agriculture in light of climate change. With this experience I am ideally prepared to disseminate our new software *rstanarm* and *shinystan* in workshops, tutorials and online.

I am personally thrilled about this project not only because of the interdisciplinary collaboration with applied clinical data scientist to port our advanced algorithms and models into data driven clinical outcomes research, but also because it will allow me to further develop some graphical theoretical approaches to improve model diagnostics in collaboration with Drs. Andreae, Betancourt and Jonah Galbry.

Bob Carpenter et al., "Stan: A Probabilistic Programming Language", (*accepted to the Journal of Statistical Software*)

B. Positions and Honors

2010 – 2013, Postdoc, Applied Statistics Center, Columbia University, New York, NY
2013 – present, Lecturer in Discipline in Political Science, Columbia University, New York, NY

C. Contribution to Science

Stan: A probabilistic programming language

I have been a core developer of Stan since 2011, which is a suite of software that enables scientists to analyze their data and make Bayesian inferences. It is difficult to estimate how many people use Stan because it is available to download for free on a variety of sites around the internet that do not directly count the number of unique users. We do know that a recent version of the Stan user manual has been downloaded over 5000 times and that the Python interface to Stan has been downloaded 6000 times in the past month. The R interface to Stan, which I am the maintainer of, is more popular than the Python interface and probably all other Stan interfaces combined.

My contribution to the Stan project has primarily been coding for multivariate probability distributions, maintaining the R packages that provide interfaces to Stan, and answering questions on the Stan-users email list, which has over 1500 members. I have participated in more than 660 threads on Stan-users and an additional 700 threads on the email list for Stan developers. Multivariate statistics is much more challenging than univariate statistics, but Stan has done more than almost any other software to provide attractive options for scientists who need to utilize multivariate probability distributions, which arise when modeling multiple outcome variables simultaneously, when employing hierarchical models, and other research contexts. In particular, Stan has done more to popularize modeling with correlation matrices than have the original papers whose ideas were incorporated into Stan and extended upon by me personally.

Some have said that it is the users of Stan who make novel contributions to science. While that is true, Stan is a necessary part of that process, even if Stan developers are not among the authors of the paper or any mention that Stan was used is relegated to a footnote. Most Stan users initially

turn to Stan when their research problem is too complicated for more widely-known software to handle well. Thus, Stan provides the infrastructure for scientists to make well-founded inferences from data.

Bob Carpenter et al., "Stan: A Probabilistic Programming Language", (*accepted to the Journal of Statistical Software*)

rstanarm: An R package to democratize Stan

Part of the power of Stan is that researchers can specify almost any model with the Stan language, and Stan's algorithms will, perhaps with a few tweaks, likely be able to efficiently produce a set of random draws from the posterior distribution of that model. However, the hurdle of specifying a model in the Stan language can be too high for beginners or for quantitative researchers that have limited formal training in probability theory. The rstanarm R package is our latest attempt to make Stan accessable to a wider set of researchers by providing a handful of models in the Stan language that are ready to be executed by R functions whose syntax is very familiar to anyone who has fit models in R before.

For example, the rstanarm package currently includes Bayesian versions of linear regression and ANOVA models, generalized linear models with a variety of likelihood and link functions, so-called generalized "mixed" models possibly with smooth additive terms such as splines, and models for ordinal outcomes. Within two days of rstanarm's first release, an R enthusiast named Wayne Folta (who is not part of the Stan development team and was until that time unkown to us) blogged that the "third reason [why everyone isn't using Bayesian methods for regression modeling] has recently been shattered in the R world by not one but two packages: brms and rstanarm. Interestingly, both of these packages are elegant front ends to Stan". Folta summarized by saying

> Bayesian modeling is a general machine that can model any kind of regression you can think of. Until recently, if you wanted to take advantage of this general machinery, you'd have to learn a general tool and its language. If you simply wanted to use Bayesian methods, you were often forced to use very-specialized functions that weren't flexible. With the advent of brms and rstanarm, R users can now use extremely flexible functions from within the familiar and powerful R framework. Perhaps we won't all become Bayesians now, but we now have significantly fewer excuses for not doing so. This is very exciting!

In the future, we would like to add to the collection of models in rstanarm and include them in other interfaces to Stan besides R. In particular, we would like to add functionality that is essential to medical researchers.

Gabry, Jonah and Ben Goodrich. "How to Use the rstanarm Package." R package vignette (2015).

mi: An R package for multiple imputation of missing data

One of the problems faced by medical researchers — as well as researchers in many other fields — is that some of the data produced by the data-generated process does not appear in the data file. In a medical context, it is often the case that some data are *uncollected*. For example, if a doctor determines that a sick patient is unlikely to have a particular medical condition, then the doctor is

less likely to order an expensive or invasive test that is presumed to turn out negative, especially if the patient is uninsured or underinsured. Conversely, patients with more extreme symptoms or more comprehensive health insurance are more likely to have the additonal test administered to them. As a result, the data file has a column for the results of this test, but its cells are empty for patients who were not tested and who differ systematically from tested patients. Thus, any analysis of the complete cases in the data file may produce severely biased estimates for the population as a whole.

The same problem manifests itself in different ways in the social sciences where some data is *missing* because the survey respondents (or other units of observation) refuse to provide it. Missing data is very common for questions about income, race, medical history, and other sensitive topics. Again, any analysis of only the complete cases is known to produce biased results.

In recent years, the most common pro-active approach to dealing with missing data is to impute the missing values multiple times using some sort of model. There are two main algorithms for modeling incomplete data, one of which is implemented by the mi R package, which has been in development for about a decade, was rewritten along with collaborators as part of my postdoctoral research at Columbia University between 2010 and 2013, and is still being maintained by me. The mi package and other implementations of similar imputation algorithms have helped a generation of scientists handle incomplete data in a more principled fashion. My fellow developers of the mi package have recently published a paper comparing its performance to the other main algorithm for modeling incomplete data and found that mi's approach is preferable for discrete variables with missingness, which are the vast majority of the variables collected in the social sciences but are common in medical research as well. For example, if a medical test is recorded simply as positive or negative when it is administered, the uncollected values on this binary variable should be imputed better by the algorithm implmented in mi than the other leading imputation algorithm.

In the future, we would like to integrate rstanarm and mi by using the former to estimate the quantities needed by mi to multiply impute the missing values. In addition, we have plans to use Stan to introduce a third category of imputation algorithm that builds on the combined strengths of the two most popular algorithms today.

Kropko, Jonathan, et al. "Multiple imputation for continuous and categorical data: Comparing joint multivariate normal and conditional approaches." *Political Analysis* 22.4 (2014): 497-519.

National Children's Study consultant

From 2013 to its ultimate demise in 2014, I was a consultant for the National Children's Study (NCS), which was being run out of the Eunice Kennedy Shriver National Institute of Child Health and Human Development at the National Institutes of Health. Although the NCS failed, I nevertheless believe that my three fellow consultants and I made an important contribution to science and were poised to perhaps make a further important contribution if the NCS had been allowed to continue.

We were asked to assess a sampling design for the NCS where hospitals were sampled from a list in the first stage and delivering mothers were sampled from each selected hospital in the second stage. Many medical researchers were opposed to this sampling design because they wanted to collect prenatal data on expectant mothers. However, we were merely asked to evaluate the cost-effectiveness of this approach using computer simulations. Although I had no previous experience with hospital data and only limited experience with survey design, I was essential to the team

by programming the simulations in R. We were given access to the State Inpatient Databases for all participating states, which we filtered for baby deliveries, and then I simulated the proposed research design on this list of hospitals and mothers. We found that this research design was not particularly efficient. Due to the high-degree of socio-demographic clustering of mothers within a hospital or technically a Primary Sampling Unit (PSU), we recommended "In the absence of cost considerations (as cost data are unavailable for this project) we suggest that a sample of at least 400 PSUs (preferably more) be drawn" in order to estimate population means with conventional levels of precision. Sampling at least 400 hospitals would be quite expensive and our recommendation — which was never made public — was misquoted to the National Academy of Science's Panel on the Design of the National Children's Study and Implications for the Generalizability of Results as a recommendation of 200 to 300 PSUs. Nevertheless, Director Collins concluded that the design was not feasible for a variety of reasons and discontinued the NCS.

Although our contribution to science was a negative one, the other three consultants and I agreed that our simulations were the first time that a survey design such as this had been rigorously studied. Without them, the NCS might have been flying blind with a design that would not have served medical researchers well. In addition, I performed some additional simulations of an alternative design that would have essentially redefined the primary sampling unit as the intersection of a hospital and an income class. The preliminary results suggested that the NCS could achieve much better estimates of the population means with 200 or perhaps fewer redefined PSUs, but we were not collectively confident enough in these preliminary results to raise this alternative with Steven Hirschfeld. At the time that the NCS was discontinued, we were planning a set of follow-up simulations to better compare these survey designs using Stan, Census data, and a list of obstetricians that worked in each hospital.

Frankel, Martin, et al. Final revision of non-public memorandum submitted to Jennifer Kwan at the National Institutes of Health on 4/1/2014.

D. Research Support

Alfred P. Sloan Foundation, G-2015-13987, *Stan Community and Continuity*, PI: Andrew Gelman
Role: Contractor during the Summer of 2015

The overall goal of the Stan project is to make it easier for scientists to use Bayesian inference in their research. The Sloan Foundation grant is specifically intended to develop the community of Stan users and to ensure that it thrives over the long term. My specific role in this regard was to develop and maintain the R package that provides an interface to Stan, which is by far the most popular way of using Stan in the community.