# GARCH models in BRMS.

## Abstract

We will implement a number of General Autoregressive Conditional Heteroskedasticity (GARCH) models into the brms package. The family of GARCH model is appropriate for time series data where the variance exhibited by the error term follows an autoregressive moving average process. brms is an R package which interfaces with Stan to fit a wide range of models. We will begin by implementing a number of the family of GARCH models within Stan. The calibration of these models will be tested using simulation based calibration. This will test whether the posterior of the data generation process specified by the model(s) can be accurately sampled by the sampling algorithm implemented by Stan. We will then provide documentation describing the models and what other pre-existing modeling options with brms can be combined with these GARCH models. Lastly, I would like to provide specific “Bayesian workflow-esqe” tutorials/articles, explaining their usage.

## Technical Details

The implementation of GARCH models into brms could be considered to have the following steps:

- coding up the GARCH(p, q) model and its adaptations optimally into Stan.
- completing simulation based calibration of the respective models.
- implement these models directly into brms.
- determine what current modelling options available in brms could be used alongside the GARCH model(s).
- provide documentation describing the models and how they can be used within brms.
- create a tutorial, where we implement the family of GARCH model now included within brms
  - explain default priors, and what other prior(s) one might consider
  - explain what other modelling options can be used alongside GARCH model(s)
  - graphically show prior and posterior predictive checking, and show how one might make decisions to how to select a model from the family of GARCH models one might want to consider.

If we assume that the variance of the errors with a time series sequence are modelled by an autoregressive moving average (ARMA), then the model is considered a generalised autoregressive conditional heteroskedasticity (GARCH) model.

The example formulation of the GARCH(p, q) model is

$$y_t = f(x_t) + \epsilon_t,\ \epsilon_t \sim N(0, \sigma_t^2),\ \sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-1}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-1}^2.$$

Some other models that exist within the family of GARCH models include the nonlinear asymmetric GARCH (NAGARCH) and the exponential GARCH (EGARCH) models.

The example formulation of the NAGARCH(1, 1) model is

$$\sigma_t^2 = \omega + \alpha(\epsilon_{t-1} - \theta\sigma_{t-1})^2 + \beta\sigma_{t-1}^2,$$

with the constraints that $\alpha \geq 0, \beta \geq 0, \omega > 0$ and $\alpha(1 + \theta^2) + \beta < 1$. This is to ensure that the variance process exhibits stationarity and is non-negative. If the parameter $\theta$ is estimated to be positive, the model captures the idea that if the response of the time series decreases, then the variance of the future terms $\epsilon_{t+1}, \epsilon_{t+2}, ...$, increase by a larger amount than if the response of the time series increases by the same magnitude.

The example formulation of the EGARCH(p, q) model is

$$log\sigma_t^2 = \omega + \sum_{i=1}^{q} \beta_i g(Z_{t-i}) + \sum_{i=1}^{p} \alpha_i log\sigma_{t-1}^2,$$

where $g(Z_t) = \theta Z_t + \lambda(|Z_t| - E(|Z_t|))$, and $\sigma_t^2$ is the conditional variance, and $Z_t$ can come from a generalised error distribution. Similar to the NGARCH model above, the EGARCH model captures the idea that a decrease of the response at $t - 1$ could have a stronger effect on the variance at time $t$, than if the response had an increase at $t - 1$ of the same magnitude.

Another possible adaption of the basic GARCH(p, q) specified above is to relax the assumption that the error terms $\epsilon_t$ is normally distributed. Although the normal distribution offers supreme convenience, it has been consistently shown that in financial time series, returns exhibit heavier tails than that expected by the normal distribution. This has led to the Student-t distribution being proposed as a possible distribution for the error terms $\epsilon_t$, in order to account for the heavier tails.

In order to test that the family of GARCH models coded in Stan are well calibrated, we will use a technique called simulation based calibration (SBC). If we define a generative model in Stan, we understand the true data generating process. We can then use SBC to check if the posterior provided by the sampler is as expected. If we simulate data from the true data generating process, then inference with respect to the model must be calibrated as we are carrying out inference using data generated from the model.

We must then implement the models directly in brms, provide technical documentation to show the model(s) have been implemented, and learn what modelling options presently available in the brms package can be used in conjunction with the GARCH model(s). I must admit that at this point in time, I am not sure how to do this stage, but I am very excited to get my hands dirty in this area, as I think that this is the area of the project where ai will learn the most about the technicalities of time series models and the effort and thought that goes into developing open source software.

## Schedule of Deliverables

### Community Bonding Period

By this stage, I will have already completed some beginner-level PR's to Stan. During this community bonding period, I will:

- contribute to issues being raised on the Stan Discourse regarding time series modelling within BRMS.
- gain a thorough understanding of the structure of Stan code being generated under the hood by the pre-existing time series models.
  - discuss with mentors choices made within the Stan code under the BRMS function calls to ensure I implement optimised Stan code.
- make sure the blog is set up, and promote the beginning of the project, and inform people of what the expected results of the project will be.

### Phase 1

#### Week 1 (7th June - 13th June)

- Implement GARCH(p, q) in Stan, allow for autoregressive distributional parameters for the mean component also.
- Carry out simulation based calibration for the above model.
- Experiment with the use of different priors for the above model.

#### Week 2 (14th June - 21st June)

- Discuss work done with mentors regarding basic GARCH(p, q).
- Either work on suggested changes or begin implementing into BRMS.

#### Week 3 (22nd June - 28th June)

- Continue model implementation into BRMS
- Do a blog post updating how the project is going.
- Begin writing documentation for the model's use within BRMS.

#### Week 4 (29th June - 4th July)

- Work out what pre-existing models within BRMS could currently be used with the GARCH(p, q) implementation.

- Decide with mentor(s) why / why not (sadly inevitable) some modelling options cannot be combined with current GARCH(p, q) implementation in BRMS
    - Decide what modelling options are within reach to be able to be combined with the GARCH(p, q) implementation.

### Week 5 (5th July - 11th July)

- Write a second blog post indicating how the project is going.
- Take review from the mentors and other community members.
- Work on suggested changes by mentors.

### Week 6 (12th July - 18th July)

- Submit GSoC Phase 1 evaluation

## Phase 2

### Week 7 (19th July - 25th July)

- Begin implementing the wider family of GARCH models into brms.
- Review feedback reviewed in Phase 1 of the project.

### Week 8 (26th July - 1st August)

- Begin writing tutorials and provide examples on how to use GARCH models now implemented.
- Continue implementing the wider family of GARCH models into brms.

### Week 9 (2nd August - 9th August)

- Write the third blog post on how the project is going.
- Continue implementing the wider family of GARCH models into brms.
- Make tutorials on how to use GARCH models available to the community and take feedback.

### Week 10 (10th August - 16th August)

- Continue implementing a wider family of GARCH models into brms.
- Note that if time permits, other things I would like to do would be to:
    - do a simulation based calibration case study for the wider family of GARCH models
    - implement GARCH models where the error term $\epsilon_t$ can be Student-t distributed.

## Final Week

- Submit GSoC final evaluations.

- Write project summary as final blog post

## Development Experience

I have not contributed to any open source projects before, although I would like this to be a major part of my graduate studies in the near future. I have code from my Honours on the effect of modelling fake news on elections available on my GitHub (please do not look, I am embarrassed of how bad it was, it used JAGS :( ). I have contributed code (not open source) during my internship at AgResearch, where we implemented Bayesian Hierachical models in Stan and brms for the testing of parasitic resistance on New Zealand farms, and during an internship at a high-frequency trading firm, where we used Python in developing volatility models and machine learning techniques for trading strategies. Over the past four years, I have completed countless assignments using R, Python, and some Java.

## Other Experiences

I have gained exposure to Stan and BRMS during a research internship, where we built multilevel hierarchical models for the parasitic resistance shown on New Zealand farm(s). My Honours dissertation focused vaguely on Bayesian-type methods (though looking back it looks rather cringeworthy, code can be found at Conor's embarrassing thesis code) . I have completed coursework involving Bayesian inference and time series modelling to a high level (A+'s), and have used volatility models in an internship with a high frequency trading firm. Currently, I am about to begin my PhD studies focusing on Bayesian Federated Learning at the Queensland University of Technology.

## Why this project?

I love the BRMS package, and think tooling like this is an awesome step in building pieces of software that help improve Bayesian workflow immensely. I think that the ability to implement time series models that allow for auto-regressive terms to be with both the mean and variance parameters will expose the BRMS package to a much wider audience.

The Stan community has contributed greatly to my knowledge and enjoyment of Bayesian Inference. Although I have only recently registered an account on the Stan forums, lurking in the forums, including finding the helpful responses of a few of the potential mentors for this project, have helped me greatly with the modelling I have done in Stan and BRMS to date. I honestly could not consider continuing my studies in Statistics in an area that was not Bayesian.

I have a strong interest in producing open source software during my research studies. All of the mentors proposed for this project have been champions of developing open source software for the wider community for a number of years, and I would love to learn from them best practices on how to develop software that is useful and beneficial to the wider community that may access this software.

I have no work commitments over the Google Summer of Code period and my studies are entirely research focussed. Therefore, I will have plenty of time. Due to my strong interest in probabilistic programming, love of the Stan community, and the amount of time on hands, I believe that I will certainly be able to complete this

## Appendix

Please find attached below my CV and academic transcript:

CV_Transcript_Conor_Hassan.pdf

Both this proposal and my CV and academic transcript can be found at the following link:

Application repository

**Contact details:**

Email: conordhassan@gmail.com

Phone: +64 22 309 2013

GitHub: My GitHub page

Twitter: @HassanConor

Stan Discourse username: ConorHassan