1. The Motivation

Statistical inference:
- goal: gain knowledge about model from data
- typical analysis tasks:
  - estimation of model parameters
  - model comparisons
  - hypothesis testing
  - limit setting
  - goodness-of-fit tests

2. The Theory

Bayesian inference:
- use Bayes’ theorem to update knowledge about model parameters \( \lambda \) from data \( D \)
- Bayes’ theorem:
  \[
  \int P(D|\lambda) P(\lambda) d\lambda = \frac{P(D)}{P(\lambda)}
  \]

3. The Curse

Computation of posterior:
- challenging task for real-world problems → curse of dimensionality
  - advanced algorithms needed to perform high-dimensional
  - sampling of posterior
  - optimization
  - integration
  - marginalization

4. The Tool

Bayesian Analysis Toolkit - BAT

- collection of algorithms & methods for Bayesian inference
- user-defined statistical models in a multi-purpose language (no tool-specific modeling language)
- use of Markov chain Monte Carlo algorithms for sampling of posterior
- toolbox-character: choose from different methods to solve custom problems
- first version released in 2008 (written in C++, depending on ROOT)

5. The Rewrite

BAT.jl
- rewrite of BAT from scratch in the Julia programming language
- design goals:
  - faster & more performant than the predecessor version of BAT
  - less dependencies & more flexible design
  - easier usage of multi-threading & distributed computing
  - open for further fields of research (become independent of ROOT)
- new code structure & new algorithms

6. The Language

The Julia language:
- modern general-purpose language (v. 1.0 released 2018)
- focus on high-performance numerical & scientific computing
- built-in package-management system
- large scientific community & growing ecosystem (~3000 registered packages)
- features: multiple dispatch, dynamic typing, meta-programming & macros, ...
- natively possible to call code in other languages, e.g. python, C, Fortran, ...

7. The Features

Current features of BAT.jl:
- registered Julia package (v. 0.5)
- user-definable likelihoods & priors
- Metropolis-Hastings sampling algorithm
- native parallel running of Markov chains
- integrals with Adaptive Harmonic Mean Integration
- call of likelihoods from other languages (directly or via binary RPC calls)
- formatted output & plotting recipes

8. The Future Prospects

Further development of BAT.jl:
- release of v. 1.0 planned for this month
- finalizing main features & API design
- new sampling algorithms (WIP)
- support for distributed computing (WIP)
- development of examples, tutorials & templates for common analysis tasks (WIP)
- much more to come...
- stay tuned!