Programming and Computational Finance

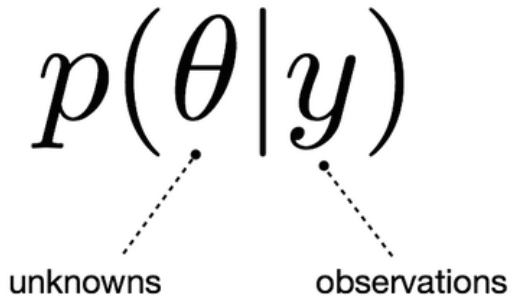
Scott Treloar

Class 7

October 2020

Bayesian Statistical Analysis

- Practical methods for making inferences from data using probability models for quantities we observe and about which we wish to learn.
- This confers several benefits to the analyst, including:
  - ease of interpretation, summarization of uncertainty
  - can incorporate uncertainty in parent parameters
  - easy to calculate summary statistics



Computational Methods in Bayesian Analysis

Step 1: Specify a probability model

- As was noted above, Bayesian statistics involves using probability models to solve problems. So, the first task is to completely specify the model in terms of probability distributions. This includes everything: unknown parameters, data, covariates, missing data, predictions. All must be assigned some probability density.

- This step involves making choices.
  - what is the form of the sampling distribution of the data?
  - what form best describes our uncertainty in the unknown parameters?

Computational Methods in Bayesian Analysis

Step 2: Calculate a posterior distribution

- The mathematical form $p(\theta|y)$ that we associated with the Bayesian approach is referred to as a posterior distribution.
- Why posterior? Because it tells us what we know about the unknown θ after having observed y.
- This posterior distribution is formulated as a function of the probability model that was specified in Step 1. Usually, we can write it down but we cannot calculate it analytically.
- But, once the posterior distribution is calculated, you get a lot for free:
  - point estimates
  - credible intervals
  - quantiles
  - predictions

Computational Methods in Bayesian Analysis

Step 3: Check your model

- Though frequently ignored in practice, it is critical that the model and its outputs be assessed before using the outputs for inference. Models are specified based on assumptions that are largely unverifiable, so the least we can do is examine the output in detail, relative to the specified model and the data that were used to fit the model.
- Specifically, we must ask:
  - does the model fit data?
  - are the conclusions reasonable?
  - are the outputs sensitive to changes in model structure?

Go to the notebook!

— Class_7_0_getting_started.ipynb

— Class_7_1_linear regression.ipynb

Probabilistic Programming

Blackbox Machine Learning

Train classifier to distinguish between distributions.

Classification framework **not well suited**.

Blackbox: Not good at conveying what was learned.

Frequentist Statistics

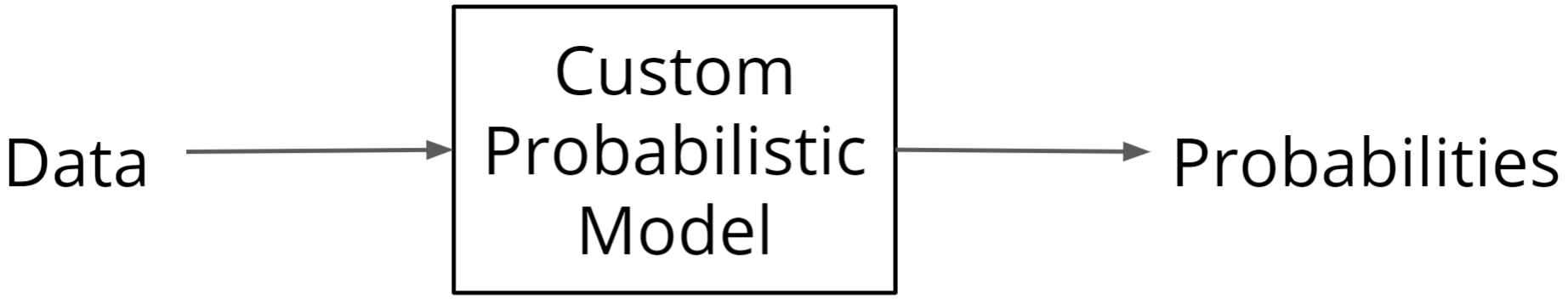
Gray box - **implicit assumptions**:

- normality (violated here, daily returns have **heavy tails**)

Hard to change assumptions.

p-value is **not** the probability we care about.

Probabilistic Programming

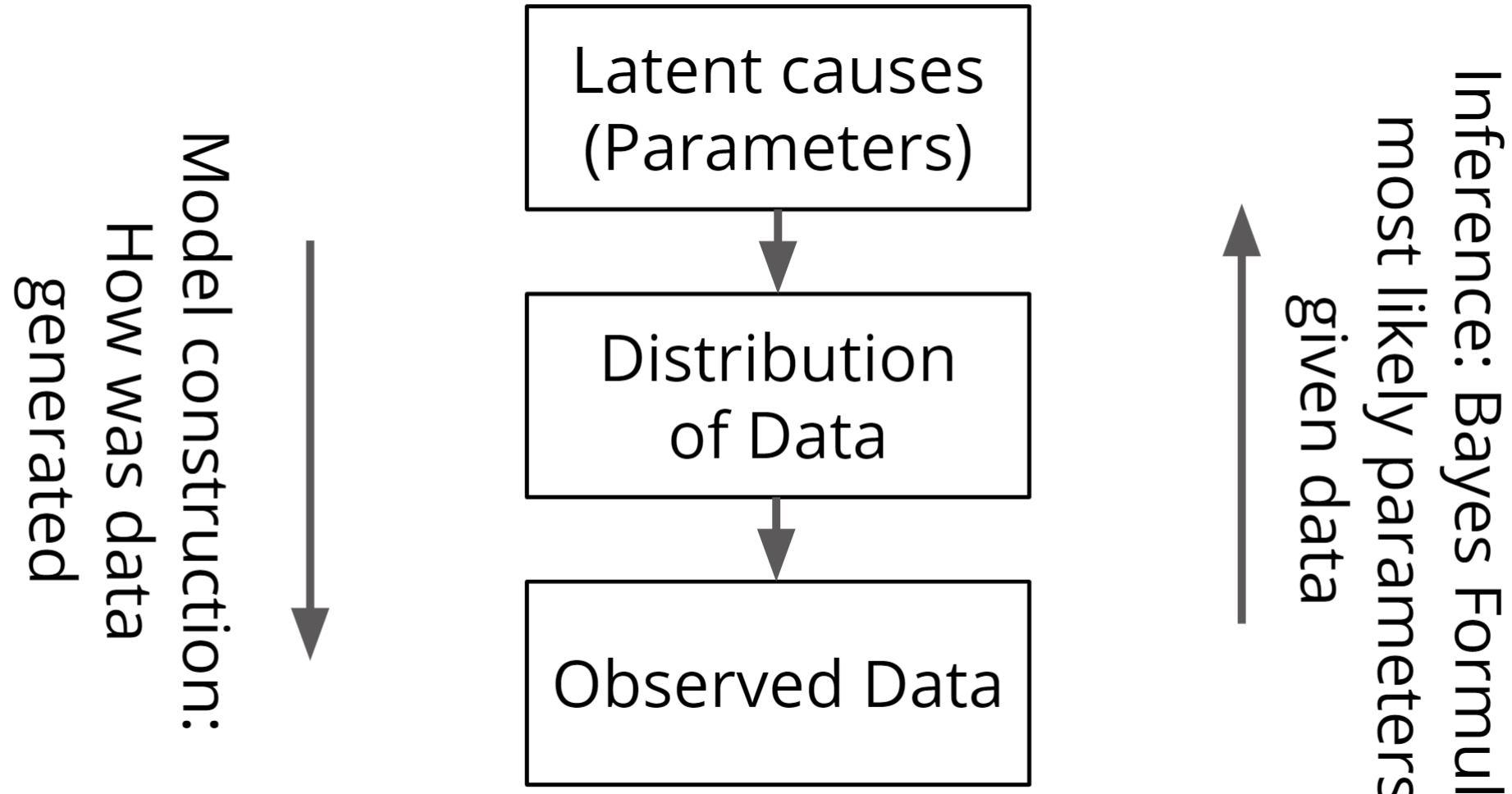


Build **custom model** in **code** (open box)

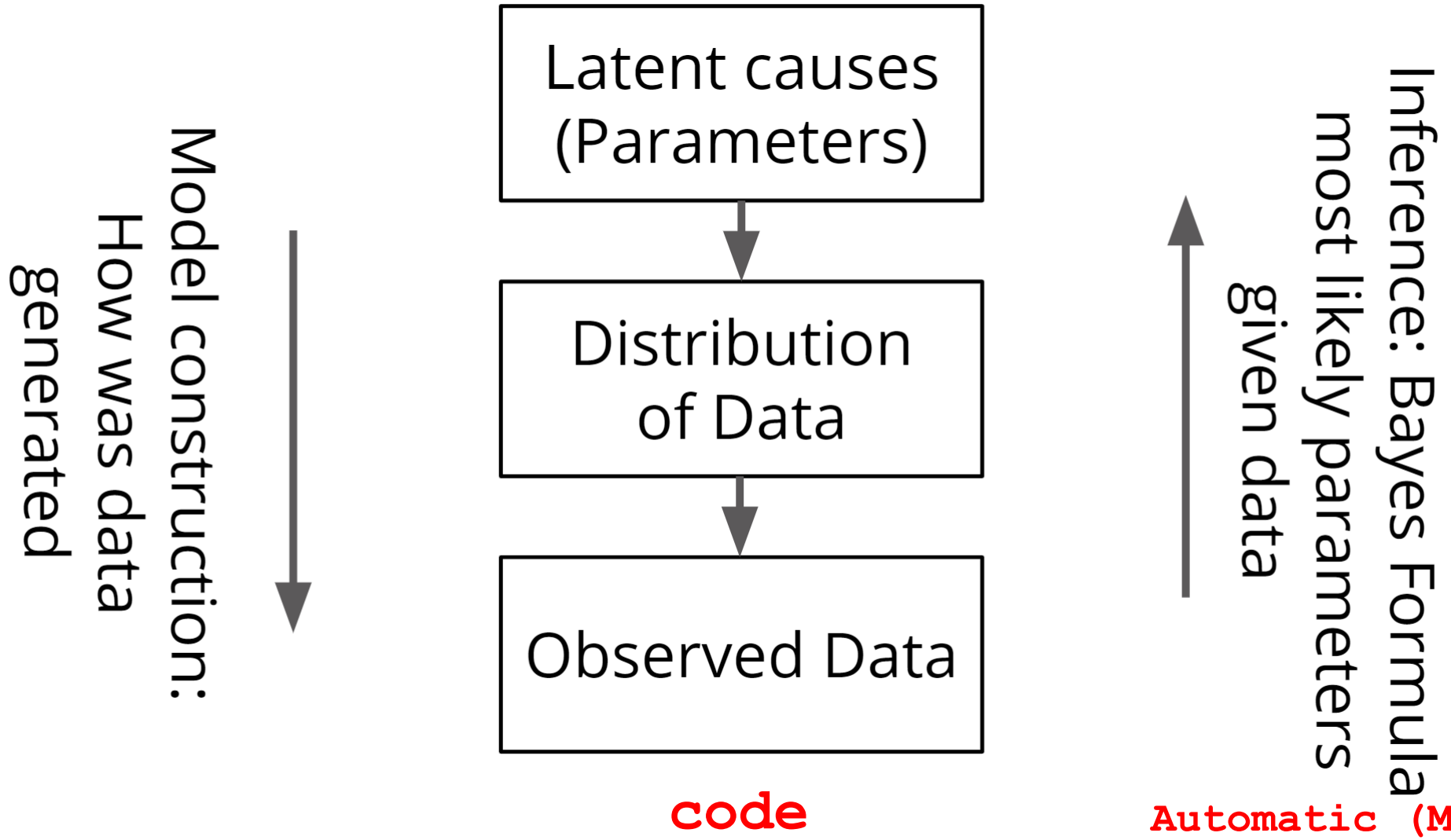
Automatic inference with MCMC

Bayesian (actual probabilities, priors)

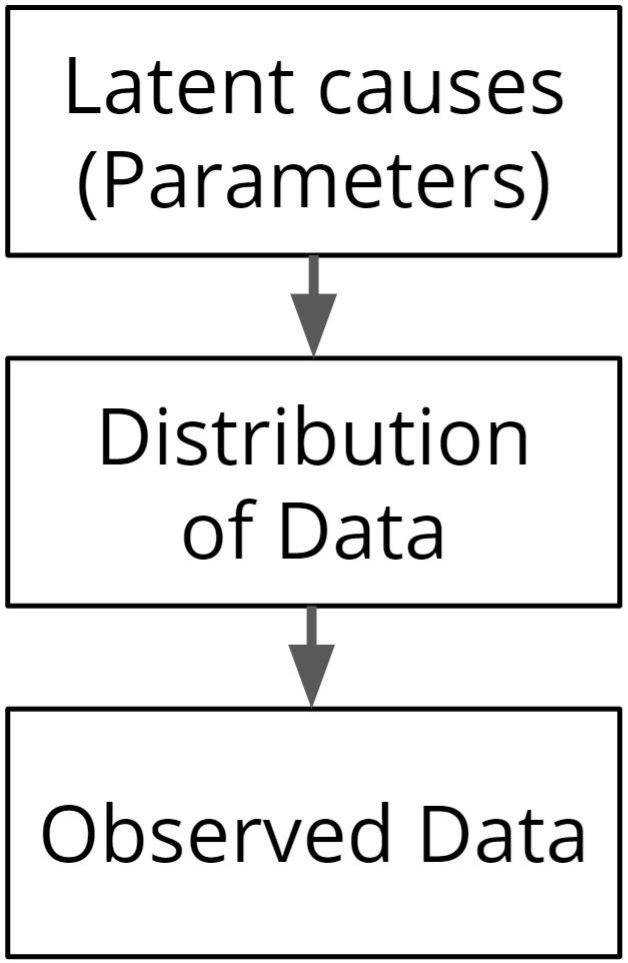
Bayesian Modelling

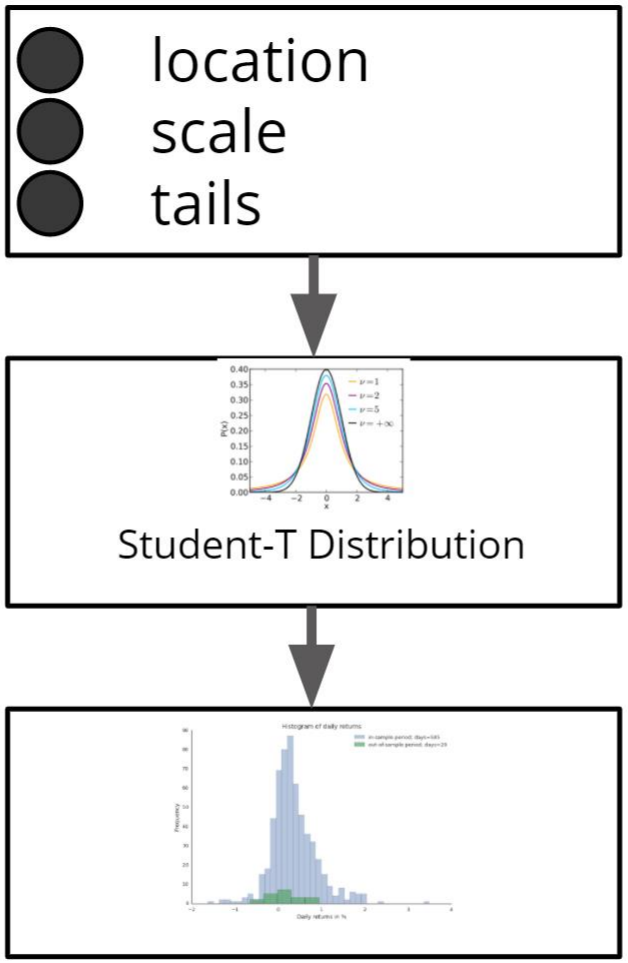


Probabilistic Programming



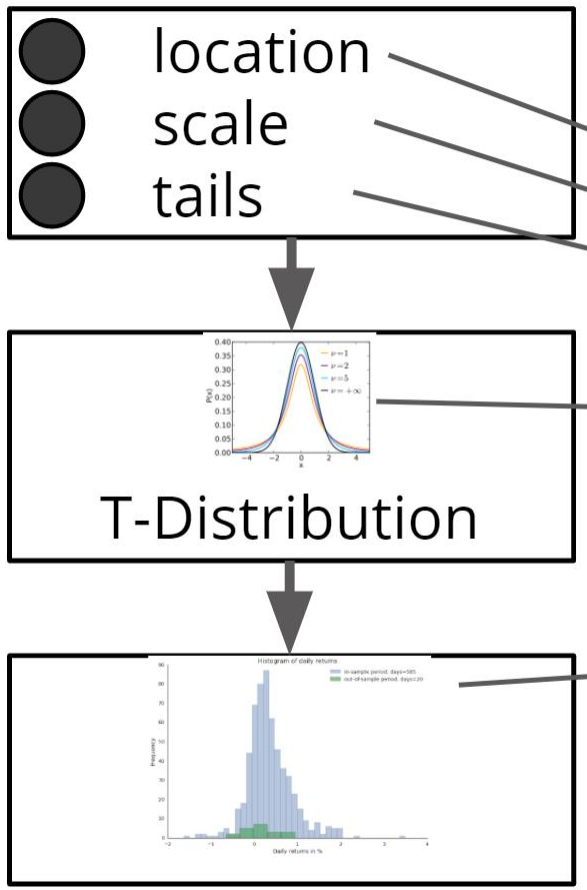
Modeling financial returns





Inference: Bayes Formula
probability of parameters
given data

Probabilistic Programming with PyMC3

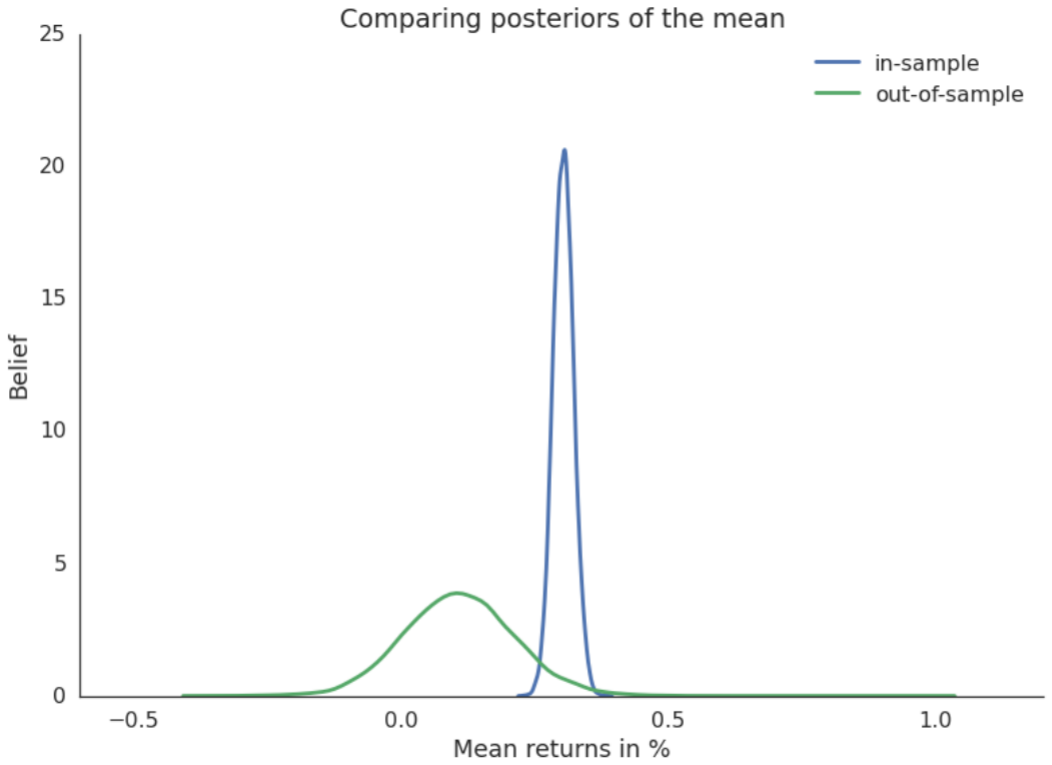


Model specification

```
from pymc3 import *

with Model() as model:
    # Parameters with priors
    location = Uniform('location', lower=-1, upper=1)
    scale    = Uniform('scale', lower=0, upper=1.)
    tail     = Uniform('tail', lower=1, upper=20)
```

Comparing mean returns IS and OOS



Go to the notebook!

— Class_7_2_stochastic_volatility.ipynb

— Class_7_3_ar.ipynb