An Introduction to Bayesian Statistics

Mattias Villani

Department of Statistics
Stockholm University

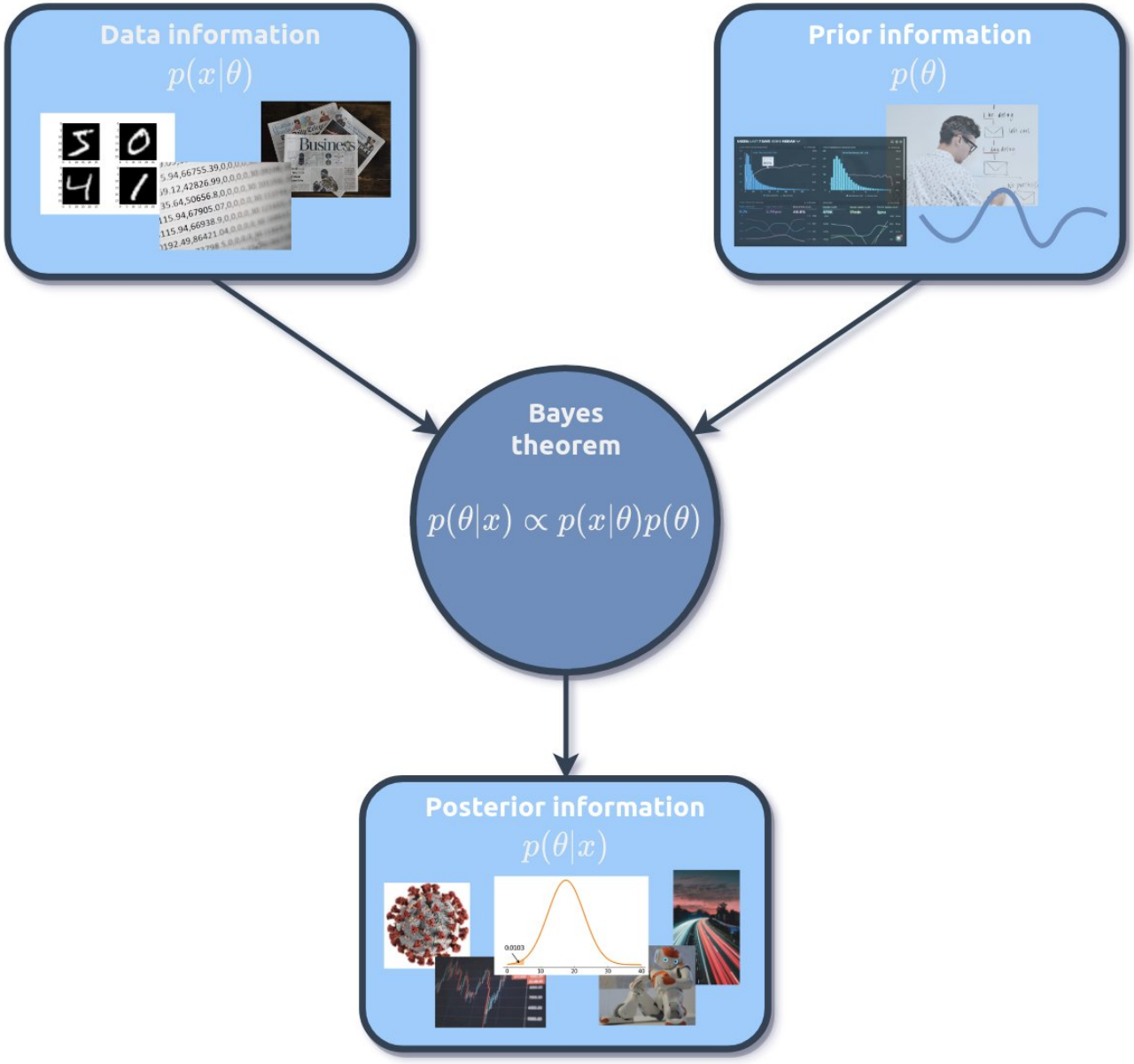
mattiasvillani.com @matvil @matvil mattiasvillani

Overview

- **The big picture**
- **Bayesian updating from data**
- **Bayesian prediction**
- **Bayesian decision making**
- **Bayesian computation and software**

- **Slides**: http://mattiasvillani.com/news.

Bayes combines data with other information sources



Bayes - the sales pitch

- Makes it possible to **combine data with other information**.
- **Predictive distribution** includes all sources of uncertainty:
  - Population noise
  - Parameter uncertainty
  - Model uncertainty
- Natural **decision making under uncertainty**.
- Natural handling of nuisance parameters by **marginalization**.
- **Inference is conditional** on the data. Early stopping.
- **Regularization** to avoid overfitting is built in.
- **Probabilistic programming languages** for practical work.
- Works without any data. **Experiment design**.

Me and Bayes - some applications

- Monetary policy - prediction and decision

- Whole-brain fMRI - activity and connectivity

- Drone search for injured people

- Probabilistic forecasts in public transport

- Coral bleaching from marine heatwaves

- The evolution of airline networks

Am I really getting my 20Mbit/sec?

- I have a 50Mbit/sec internet connection.
- My internet provider promises at least 20Mbit/sec on average.
- **Data**: $\mathrm{x} = (15.77, 20.5, 8.26, 14.37, 21.09)$ Mbit/sec.
- **Measurement errors**: $\sigma = 5$ (± 10Mbit with 95% probability)
- **Data model**

$$X_1, ..., X_n | \theta \overset{\text{iid}}{\sim} \mathrm{N}(\theta, \sigma^2)$$

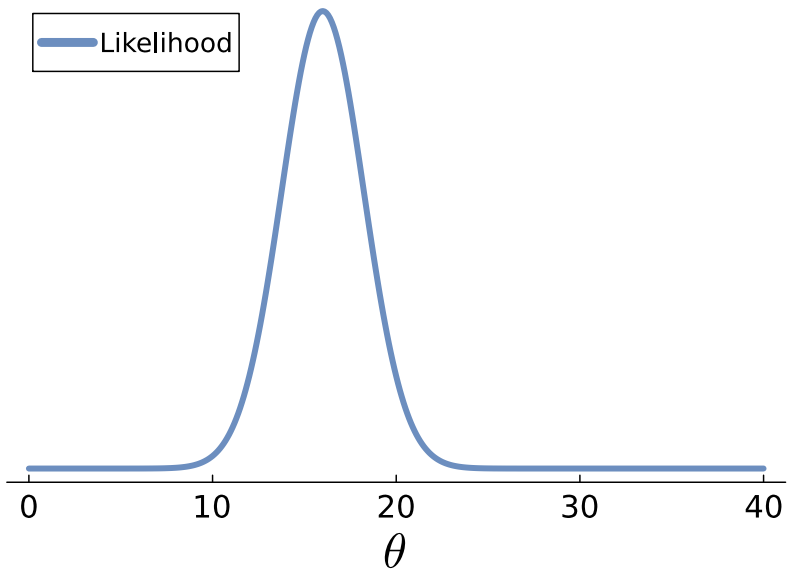
Likelihood function

Likelihood function

$$X_1, ..., X_n | \theta \overset{\text{iid}}{\sim} \mathrm{N}(\theta, \sigma^2)$$

viewed as a function of θ, for **observed data** $x_1, ..., x_n$.

- The likelihood is **proportional** to a $\mathrm{N}(\bar{x}, \sigma^2/n)$ density.

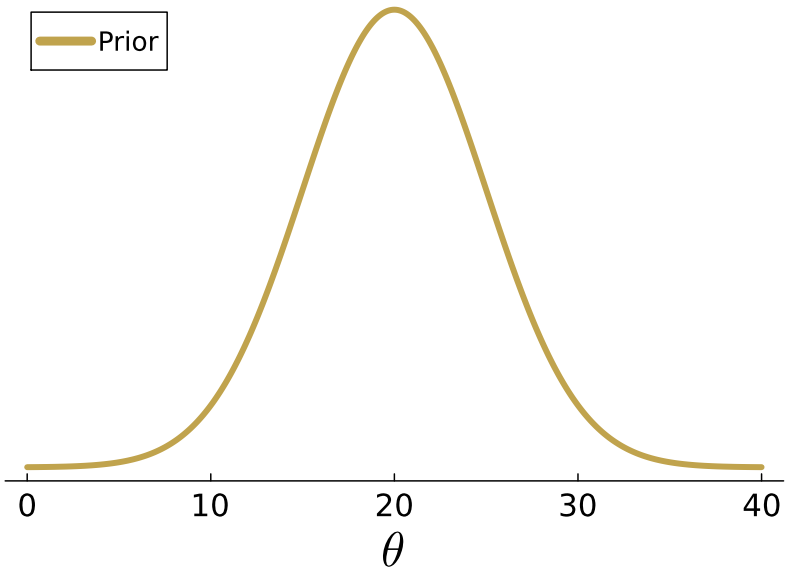


- The likelihood is **not** a probability density for θ. 😢
- But wait, how **could** it be? θ is not random! 🤔

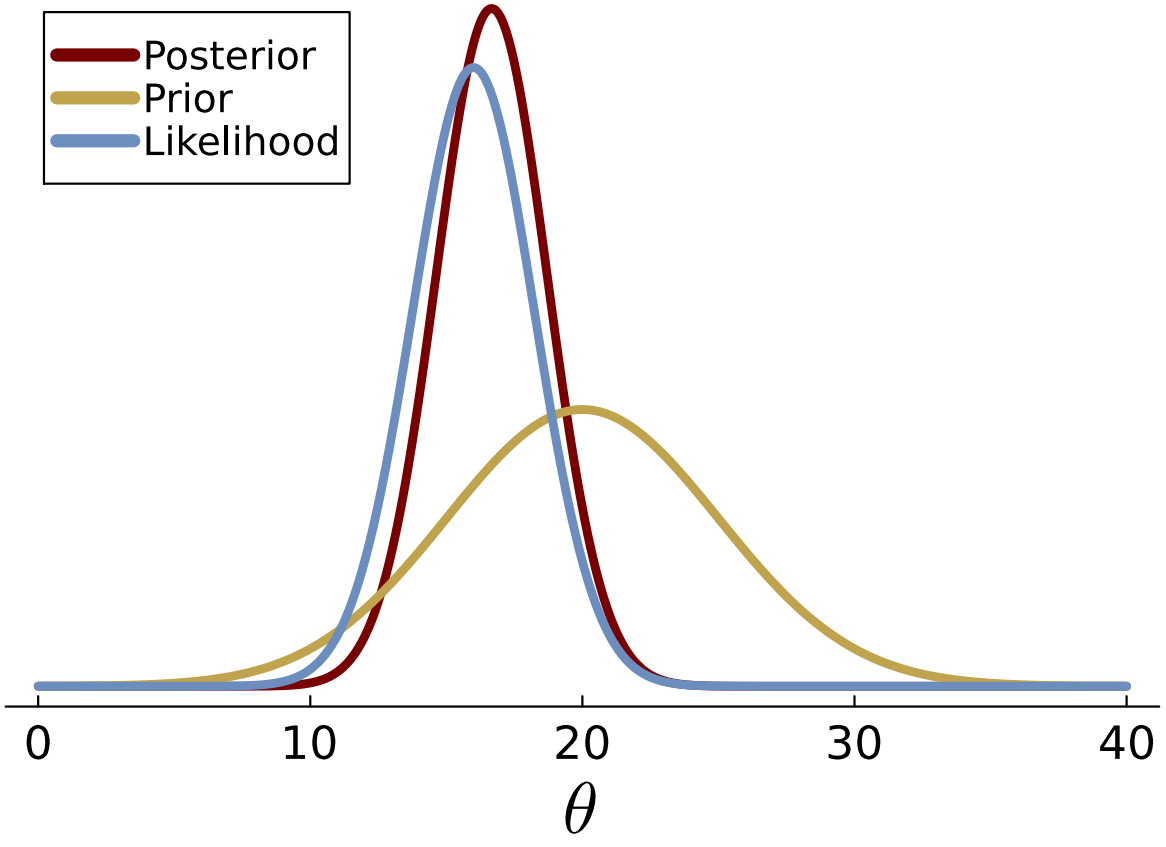
Subjective probability and Bayes!

- Bayesian learning is based on **subjective probability**.
- Probability as subjective **degrees of belief**.
- All **unknowns** should be quantified by subjective probability.
- 👱 "$\Pr(10\text{th decimal of } \pi \text{ is } 5) = 0.1$"
- 🤓 "$\Pr(10\text{th decimal of } \pi \text{ is } 5) = 1.0$"
- **Prior distribution**

$$\theta \sim \mathrm{N}(\mu_0, \tau_0^2)$$



Bayes combines data and prior information

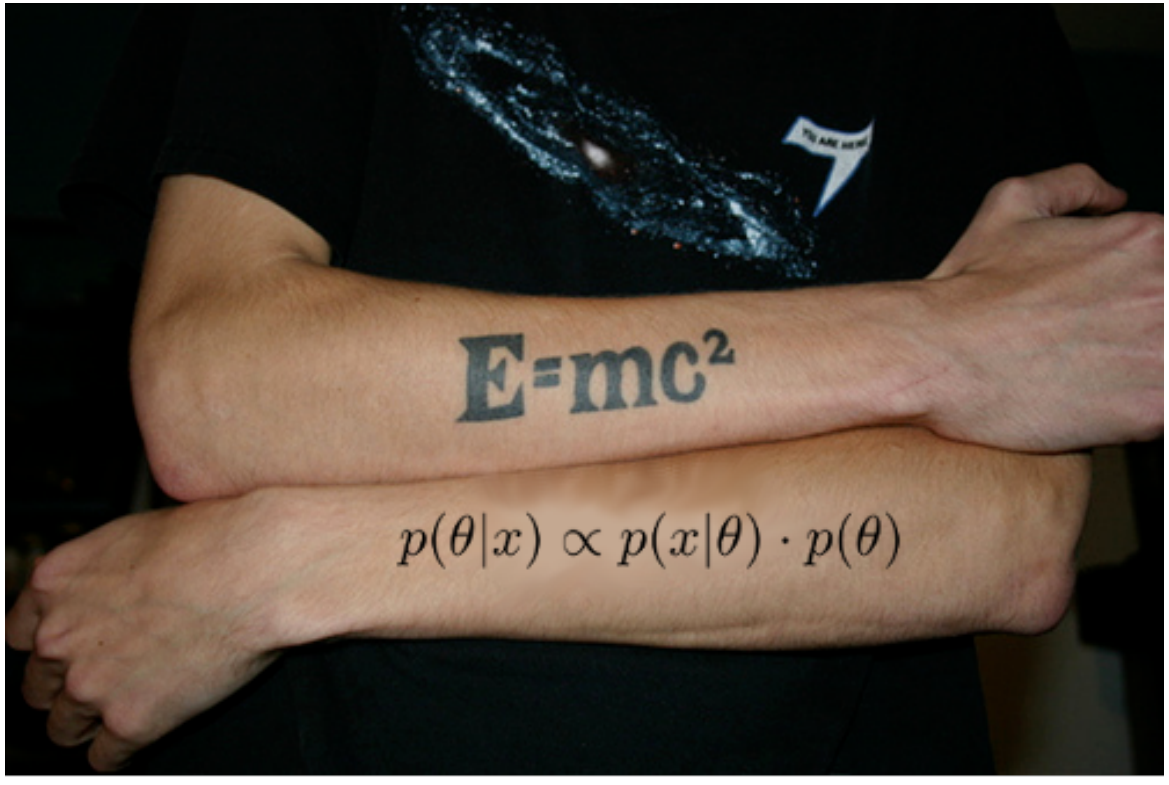


Great theorems make great tattoos

- Data and prior is combined through **Bayes' theorem**

$$p(\theta|x_1, ..., x_n) \propto p(x_1, ..., x_n|\theta) \times p(\theta)$$

Posterior $\propto$ **Likelihood** $\times$ **Prior**



Normal data, known variance - normal prior

Posterior distribution

$$\theta|x_1, ..., x_n \sim \mathrm{N}(\mu_n, \tau_n^2)$$

Posterior mean

$$\mu_n = w\bar{x} + (1 - w)\mu_0$$

with weight on data

$$w = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}} = \frac{\text{data precision}}{\text{data precision} + \text{prior precision}}$$

Posterior variance

$$\tau_n^2 = \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}} = \frac{1}{\text{data precision} + \text{prior precision}}$$

Interactive - Bayes for Gaussian iid model

Data:

n 1

σ 5

Observations used: $\boldsymbol{x} = (\ 15.77)$

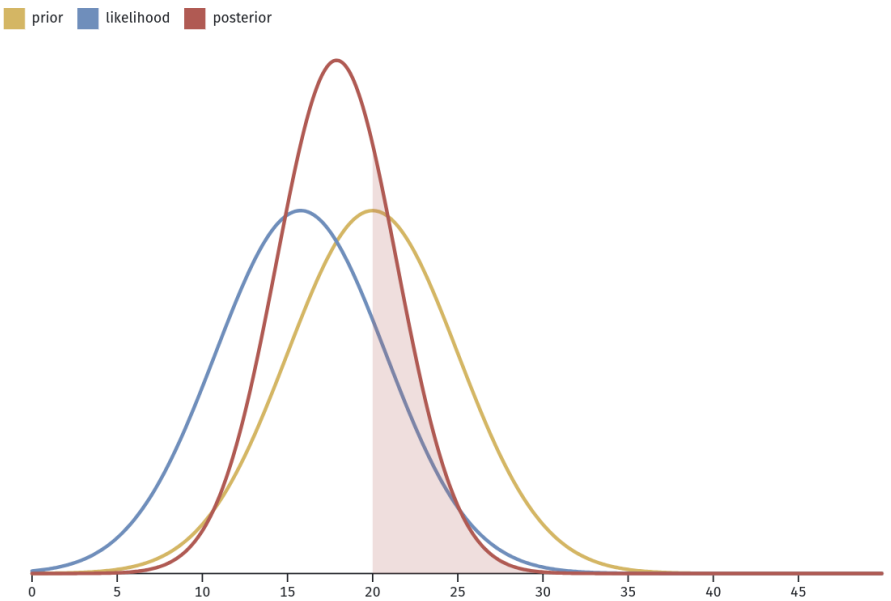
Prior:

μ_0 20

τ_0 5

quantile posterior 20

Posterior probability: $P(\theta \geq 20|\boldsymbol{x}) = 0.275$



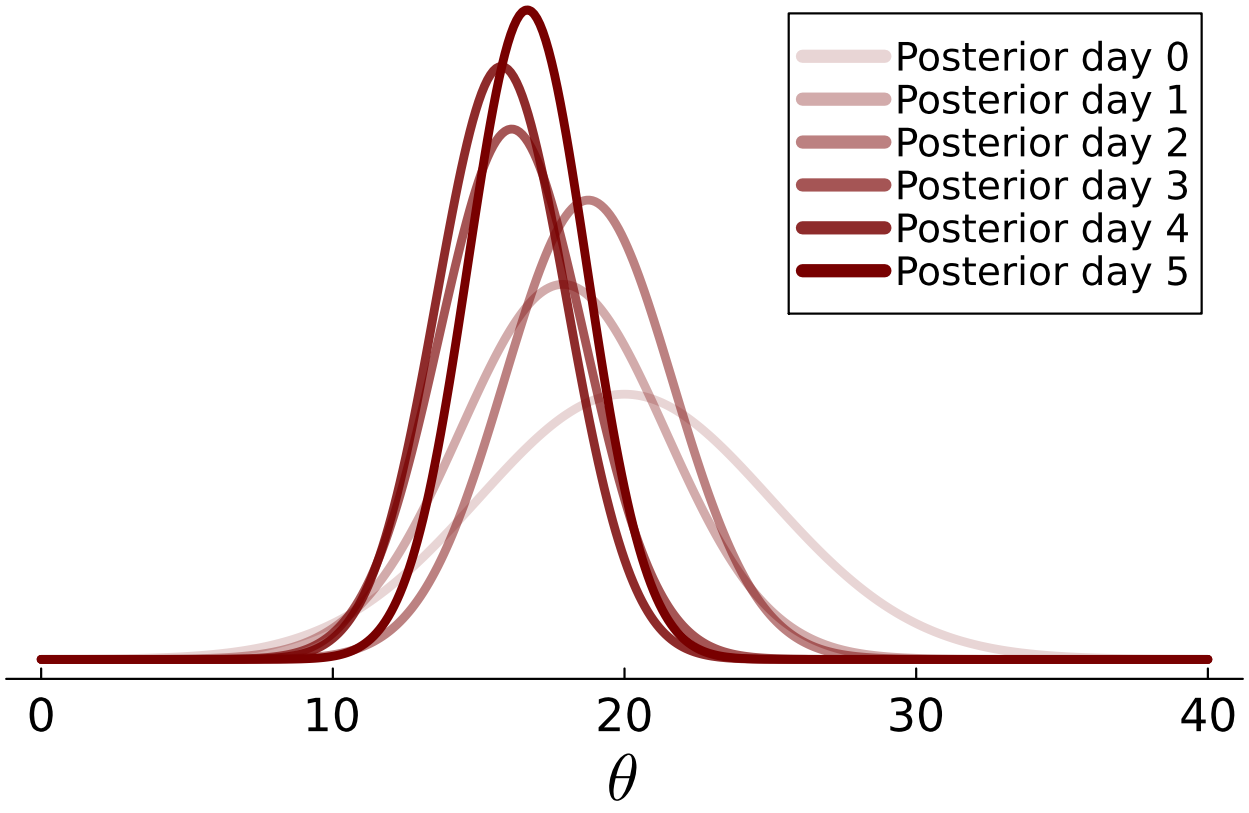
Interactive plots in Javascript

- Bernoulli model - Beta prior
- Poisson model - Gamma prior
- Exponential model - Gamma prior
- Multinomial model - Dirichlet prior
- Normal model with known variance - Normal prior
- Normal model - Normal-InvGamma prior
- Linear regression - Conjugate prior

- Collection of more Bayesian interactive widgets.

Bayesian sequential/online learning

- Yesterday's posterior is today's prior.



Bayesian Prediction

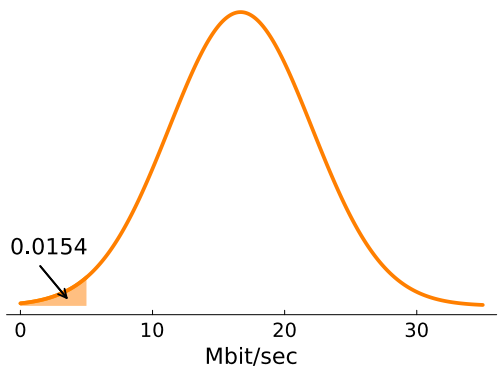
- **Predictive distribution** averages over the unknown parameter

$$\underbrace{p(x_{n+1}|x_{1:n})}_{\text{predictive dist}} = \int \underbrace{p(x_{n+1}|\theta)}_{\text{model}} \underbrace{p(\theta|x_{1:n})}_{\text{posterior}} d\theta$$

- Normal data, normal prior:

$$x_{n+1}|x_{1:n} \sim N(\mu_n, \sigma^2 + \tau_n^2)$$

- My streaming buffers whenever $x < 5$ Mbit/Sec. 



- The predictive distribution is easily computed by **simulation**.

Decision making under uncertainty

- Let $a \in \mathcal{A}$ be an **action**.
  - Example 1: treatment/no treatment
  - Example 2: central bank's interest rate
  - Example 3: screen resolution on mobile gaming
- Let θ be an **unknown quantity**.
  - Example 1: effectiveness of treatment
  - Example 2: inflation rate a year from now
  - Example 3: gamer's reaction to lower resolution
- Choosing action a when state of nature is θ gives **utility**

$$U(a, \theta)$$

Optimal Bayesian decisions

- The eternal Umbrella decision:

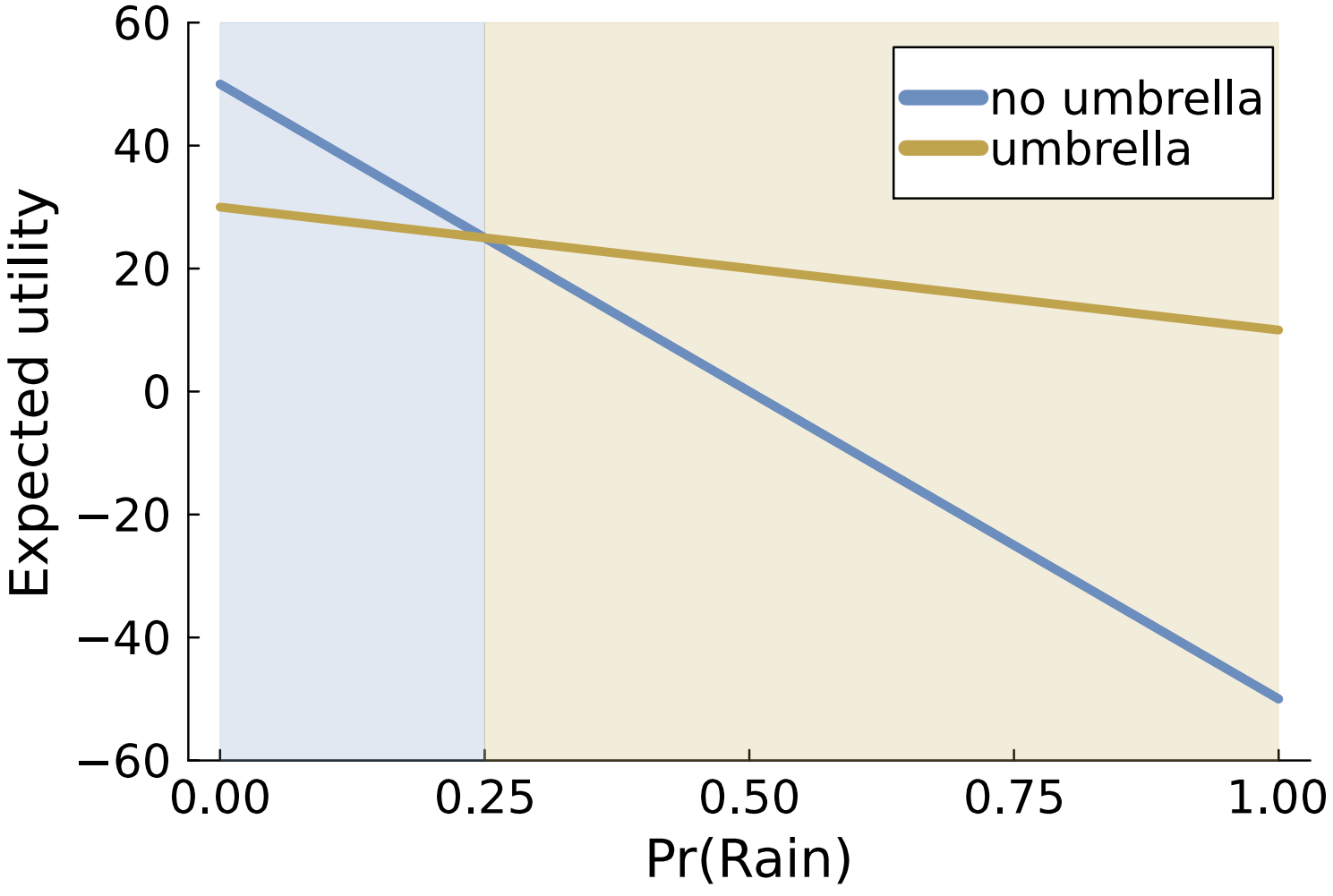
	Rain	Sun
No umbrella	-50	50
Umbrella	10	30

- Ad hoc decision rules:
  - *Minimax*. Minimizes the maximum loss.
  - *Minimax-regret* ... 😴
- **Bayesian theory**: maximize **posterior expected utility** 😍

$$E_{p(\theta|\boldsymbol{x})}[U(a, \theta)],$$

where $E_{p(\theta|\boldsymbol{x})}$ denotes the posterior expectation.

The umbrella decision



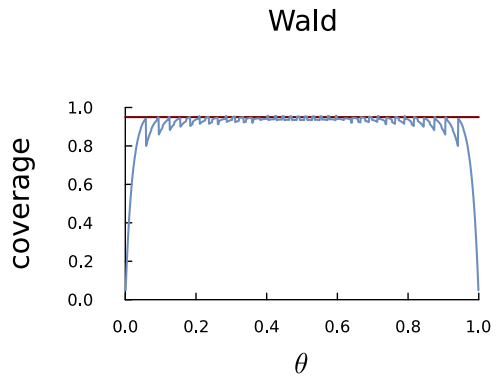
Bayesian vs Frequentist inference

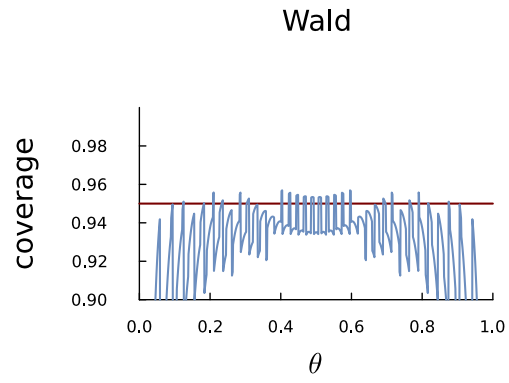
- **Frequentist** inference: confidence intervals cover in 95% of **all possible samples** etc.
- **Bayes**: credible intervals are **probability statements for θ conditional on the observed data**.
- Illustrative toy example: $X_1, \ldots, X_n | \theta \overset{\text{iid}}{\sim} \text{Uniform}(\theta, \theta + 1)$. Observed data: $x_{\max} - x_{\min} = 0.99$.

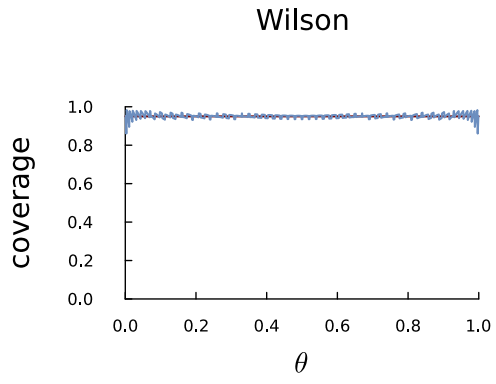
- Bayes *can* use other information via the prior.
- Bayes *has* to use a prior, but noninformative prior is an option.

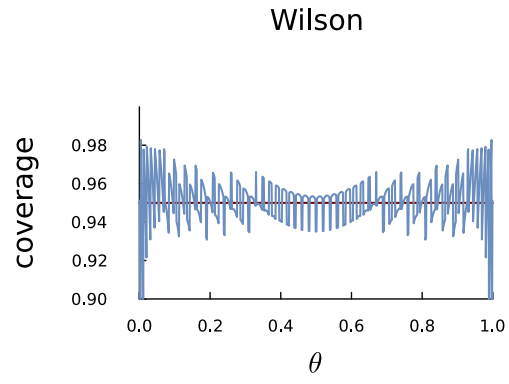
- Bayes gives **probability distribution for the unknown**.
- Point estimates and intervals possible, but there is more.
- Probability distribution for unknowns $\rightarrow$ predictive distributions, decisions, model comparison.

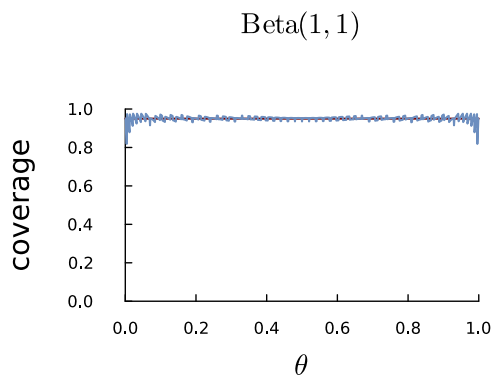
Intervals for a proportion

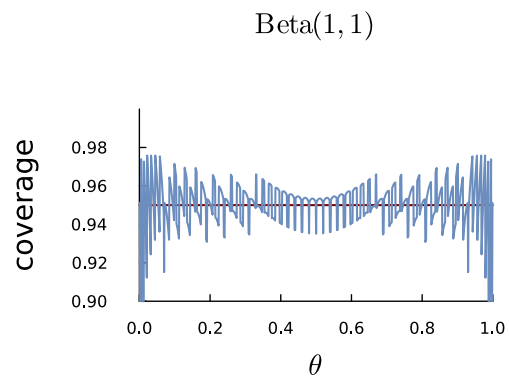










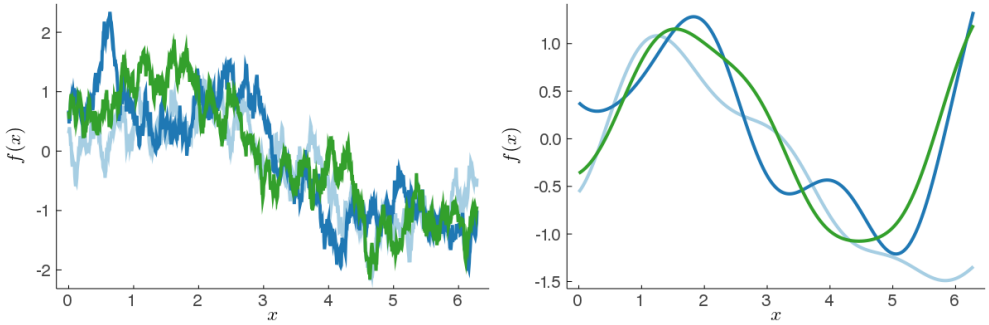


Regularization is a prior

- **Linear regression**

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \varepsilon, \quad \varepsilon \sim N(0, \sigma_\varepsilon^2)$$

- $n < p$. More predictors than observations. **Overfitting**.
- Solution: **regularization**. Ridge, Lasso etc
- **Regularization** can be viewed as a **Bayesian prior**.
- Ridge (**L2**) $\Longleftrightarrow$ Normal prior on each β_j
- Lasso (**L1**) $\Longleftrightarrow$ Laplace prior on each β_j + mode
- **Horseshoe** and other global-local priors.
- We solve the $n < p$ problem by **adding prior information**.
- **Gaussian processes**: $y = f(x) + \varepsilon$, $f(x)$ is smooth a priori.



Bayesian model comparison

- **Likelihood ratio** comparing models M_1 and M_2 on data $\mathbf{y}$

$$\frac{p(\mathbf{y}|\hat{\theta}_1, M_1)}{p(\mathbf{y}|\hat{\theta}_2, M_2)}$$

- **Posterior model probabilities**

$$\underbrace{\Pr(M_k|\mathbf{y})}_{\text{posterior model prob.}} \propto \underbrace{p(\mathbf{y}|M_k)}_{\text{marginal likelihood}} \cdot \underbrace{\Pr(M_k)}_{\text{prior model prob.}}$$

- The **marginal likelihood** for model M_k with parameters θ_k

$$\underbrace{p(\mathbf{y}|M_k)}_{} = \int p(\boldsymbol{y}|\theta_k, M_k) p(\theta_k|M_k) d\theta_k.$$

- The **Bayes factor**

$$B_{12}(\boldsymbol{y}) = \frac{p(\mathbf{y}|M_1)}{p(\mathbf{y}|M_2)}$$

- A small model nested in a larger model can 'win'.

Model choice in multivariate time series

- **Multivariate time series**

$$\mathbf{x}_t = \alpha\beta'\mathbf{z}_t + \Phi_1\mathbf{x}_{t-1} + ...\Phi_k\mathbf{x}_{t-k} + \Psi_1 + \Psi_2 t + \Psi_3 t^2 + \varepsilon_t$$

- Need to choose:
  - **Lag length**, $(k = 1, 2.., 4)$
  - **Trend model** $(s = 1, 2, ..., 5)$
  - **Long-run (cointegration) relations** $(r = 0, 1, 2, 3, 4)$.

The most probable (k, r, s) combinations in the Danish monetary data.

k	1	1	1	1	1	1	1	1	0	1
r	3	3	2	4	2	1	2	3	4	3
s	3	2	2	2	3	3	4	4	4	5
$p(k, r, s\|y, x, z)$	.106	.093	.091	.060	.059	.055	.054	.049	.040	.038

- **Models need not be nested** like in this example.

Bayesian model averaging

- Let γ be the unknown response of a patient to a treatment.
- K models $M_1, \ldots, M_K$ are entertained for estimating γ.
- **Model averaging**: average the results over all K models:

$$p(\gamma|\boldsymbol{y}) = \sum_{k=1}^{K} p(M_k|\boldsymbol{y}) p_k(\gamma|\boldsymbol{y}),$$

  where

  - $p_k(\gamma|\boldsymbol{y})$ is the posterior of γ in model M_k
  - $p(M_k|\boldsymbol{y})$ is the posterior model probability for M_k

- γ can be a future/new value. **Model averaged predictions**.
- **Bayesian variable selection** - averages over all covariate combinations. Spike-and-slab prior.

Comparing models is hard - Bayes not a silver bullet

- The prior on the parameters within each model $p(\theta|M)$ is crucial for the **marginal likelihood**

$$p(\boldsymbol{y}|M) = \int p(\boldsymbol{y}|\theta, M)p(\theta|M)d\theta.$$

- A **Bayesian model** is **likelihood + prior**.
- A marginal likelihood is an average with respect to $p(\theta|M)$. **Priors matter!**
- Not possible to use improper priors for model comparison.
- Vague priors for model comparison is a bad idea.
- **Log predictive score** is a robustified marginal likelihood.
- Bayesian model comparison can be **overconfident**.

Bayesian computations

- Posterior **sampling**:
  - **Gibbs sampling** - not always possible
  - **Markov Chain Monte Carlo** - the workhorse algorithm
  - **Hamiltonian Monte Carlo** (HMC) - MCMC for high-dimensional posteriors

- **Normal approximation**
  - Bernstein-von Mises theorem:

$$\boldsymbol{\theta}|x_1, \ldots, x_n \xrightarrow{d} N\left(\hat{\boldsymbol{\theta}}, I^{-1}(\hat{\boldsymbol{\theta}})\right) \quad \text{as } n \to \infty$$

  - Numerical optimizer + Autodiff.

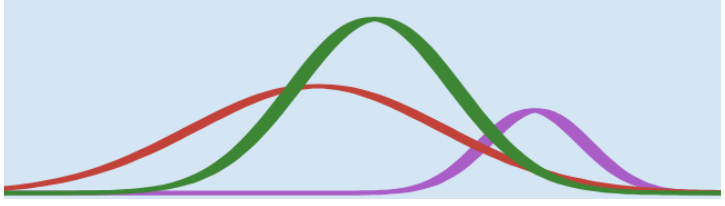
- **Variational inference**
  - approximate posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ by simpler distribution $q(\boldsymbol{\theta})$
  - find $q(\boldsymbol{\theta})$ that minimizes the Kullback-Leibler divergence:

$$\int \ln \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{x})} q(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

Probabilistic programming languages for Bayes

- **Stan** is a probabilistic programming language for Bayes based on HMC.
- C++ using the R package `rstan`. Bindings from Python.

- **Turing.jl** is a probabilistic programming language in Julia.
- Written in `Julia`, which is fast natively.



HMC sampling for iid normal model in Turing.jl

```julia
using Turing

ScaledInverseChiSq(ν,τ²) = InverseGamma(ν/2,ν*τ²/2) # Scaled Inv-χ² distribution

# Setting up the Turing model:
@model function iidnormal(x, μ₀, κ₀, ν₀, σ²₀)
    σ² ~ ScaledInverseChiSq(ν₀, σ²₀)
    θ ~ Normal(μ₀, √(σ²/κ₀))  # prior
    n = length(x)  # number of observations
    for i in 1:n
        x[i] ~ Normal(θ, √σ²) # model
    end
end

# Set up the observed data
x = [15.77,20.5,8.26,14.37,21.09]

# Set up the prior
μ₀ = 20; κ₀ = 1; ν₀ = 5; σ²₀ = 5^2

# Settings of the Hamiltonian Monte Carlo (HMC) sampler.
α = 0.8
postdraws = sample(iidnormal(x, μ₀, κ₀, ν₀, σ²₀), NUTS(α), 10000, discard_initial = 1000)
```

HMC sampling for iid normal model in rstan

```
library(rstan)

# Define the Stan model
stanModelNormal = '
// The input data is a vector y of length N.
data {
  // data
  int<lower=0> N;
  vector[N] y;
  // prior
  real mu0;
  real<lower=0> kappa0;
  real<lower=0> nu0;
  real<lower=0> sigma20;
}

// The parameters in the model
parameters {
  real theta;
  real<lower=0> sigma2;
}

model {
  sigma2 ~ scaled_inv_chi_square(nu0, sqrt(sigma20));
  theta ~ normal(mu0,sqrt(sigma2/kappa0));
  y ~ normal(theta, sqrt(sigma2));
}
'

# Set up the observed data
data <- list(N = 5, y = c(15.77, 20.5, 8.26, 14.37, 21.09))

# Set up the prior
prior <- list(mu0 = 20, kappa0 = 1, nu0 = 5, sigma20 = 5^2)

# Sample from posterior using HMC
fit <- stan(model_code = stanModelNormal, data = c(data,prior), iter = 10000 )
```

Modeling the number of bidders in eBay auctions

variable	description	data type	original range
`nbids`	number of bids	counts	$[0, 12]$
`bookvalue`	coin's book value	continuous	$[7.5, 399.5]$
`startprice`	seller's reservation price / book value	continuous	$[0, 1.702]$
`minblemish`	minor blemish	binary	$[0, 1]$
`majblemish`	major blemish	binary	$[0, 1]$
`negfeedback`	large negative feedback score	binary	$[0, 1]$
`powerseller`	large quantity seller	binary	$[0, 1]$
`verified`	verified seller on ebay	binary	$[0, 1]$
`sealed`	unopened package	binary	$[0, 1]$

Poisson regression

$$y_i | \boldsymbol{x}_i \sim \text{Poisson}(\lambda_i)$$

$$\lambda_i = \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})$$

HMC sampling for Poisson regression in Turing.jl

```julia
using Turing

# Setting up the poisson regression model
@model function poissonReg(y, X, τ)
    p = size(X,2)
    β ~ filldist(Normal(0, τ), p)  # all β_j are iid Normal(0, τ)
    λ = exp.(X*β)
    n = length(y)
    for i in 1:n
        y[i] ~ Poisson(λ[i])
    end
end

# HMC sampling from posterior of β
τ = 10    # Prior standard deviation
α = 0.70  # target acceptance probability in NUTS sampler
model = poissonReg(y, X, τ)
chain = sample(model, Turing.NUTS(α), 10000, discard_initial = 1000)
```

- Poisson regression in rstan.

... or TuringGLM.jl with R's formula syntax

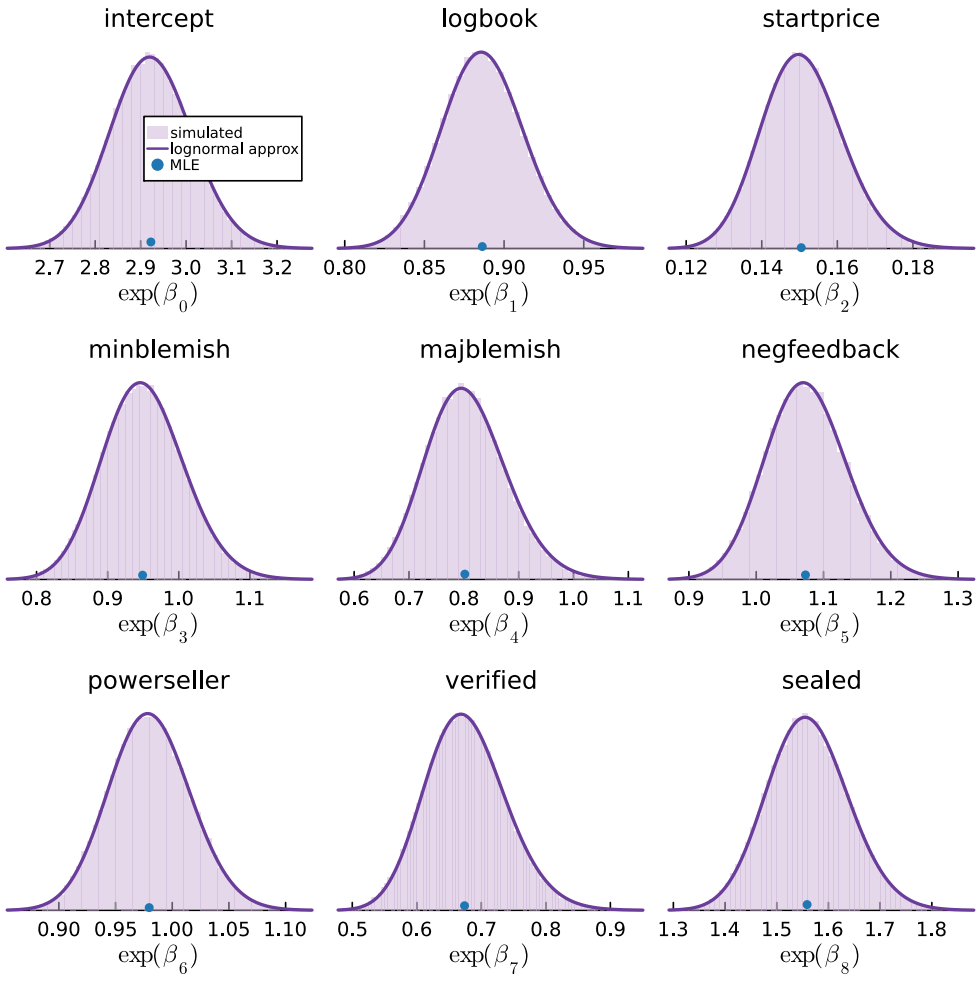
```
# Using TuringGLM.jl
using TuringGLM
fm = @formula(nbids ~ logbook + startprice + minblemish +
    majblemish + negfeedback + powerseller + verified + sealed)
model = turing_model(fm, ebay_df; model = Poisson)
chain = sample(model, NUTS(), 10000)
```

- Inspired by the brms package in R.

Marginal posteriors

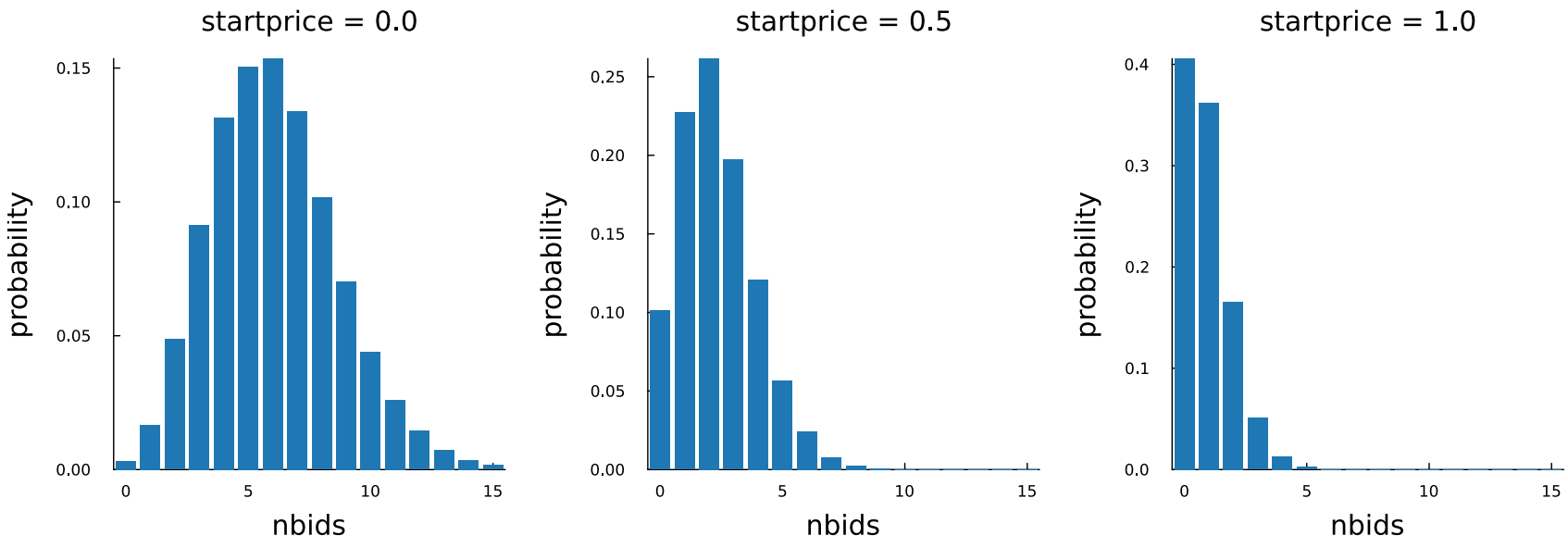
- Multiplicative model

$$E(y|\boldsymbol{x}) = \exp(\beta_0+\beta_1 x_1+\beta_2 x_2) = \exp(\beta_0)\exp(\beta_1)^{x_1}\exp(\beta_2)^{x_2}$$



Predictive distributions for different startprice

- Test auction:
  - verified, powerseller with no substantial negative feedback
  - coin with major blemish in sealed packaging
  - book value $100



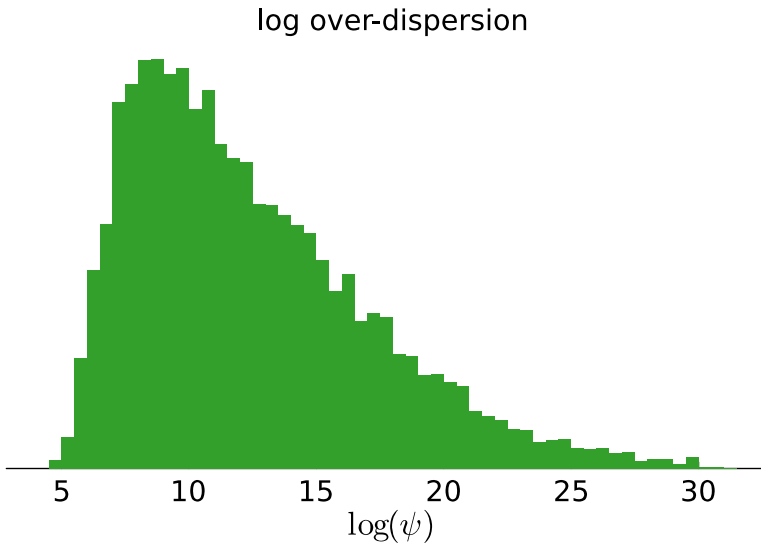
Negative binomial regression in Turing.jl

Negative binomial regression

$$y_i|\boldsymbol{x}_i \sim \text{NegBinomial}\left(\psi, p = \frac{\psi}{\psi + \lambda_i}\right), \quad \lambda_i = \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})$$

- Mean is still λ_i, but variance is larger: $\text{Var}(y_i) = \lambda_i(1 + \lambda_i/\psi)$.
- As $\psi \to \infty$ we get Poisson again.

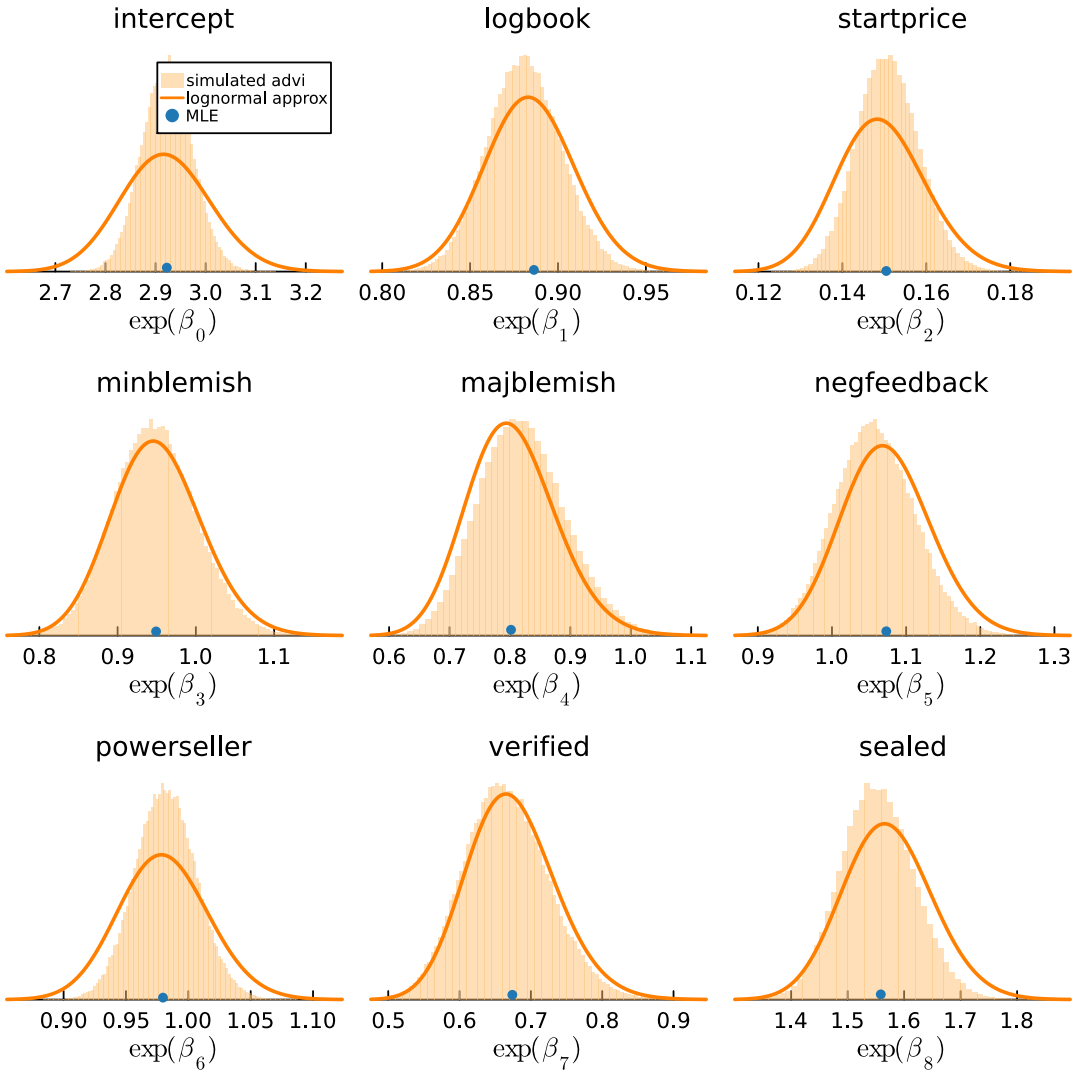
```
# Negative binomial regression
@model function negbinomialReg(y, X, τ, μ₀, σ₀)
    p = size(X,2)
    β ~ filldist(Normal(0, τ), p)
    λ = exp.(X*β)
    ψ ~ LogNormal(μ₀, σ₀)
    n = length(y)
    for i in 1:n
        y[i] ~ NegativeBinomial(ψ, ψ/(ψ + λ[i]))
    end
end
```



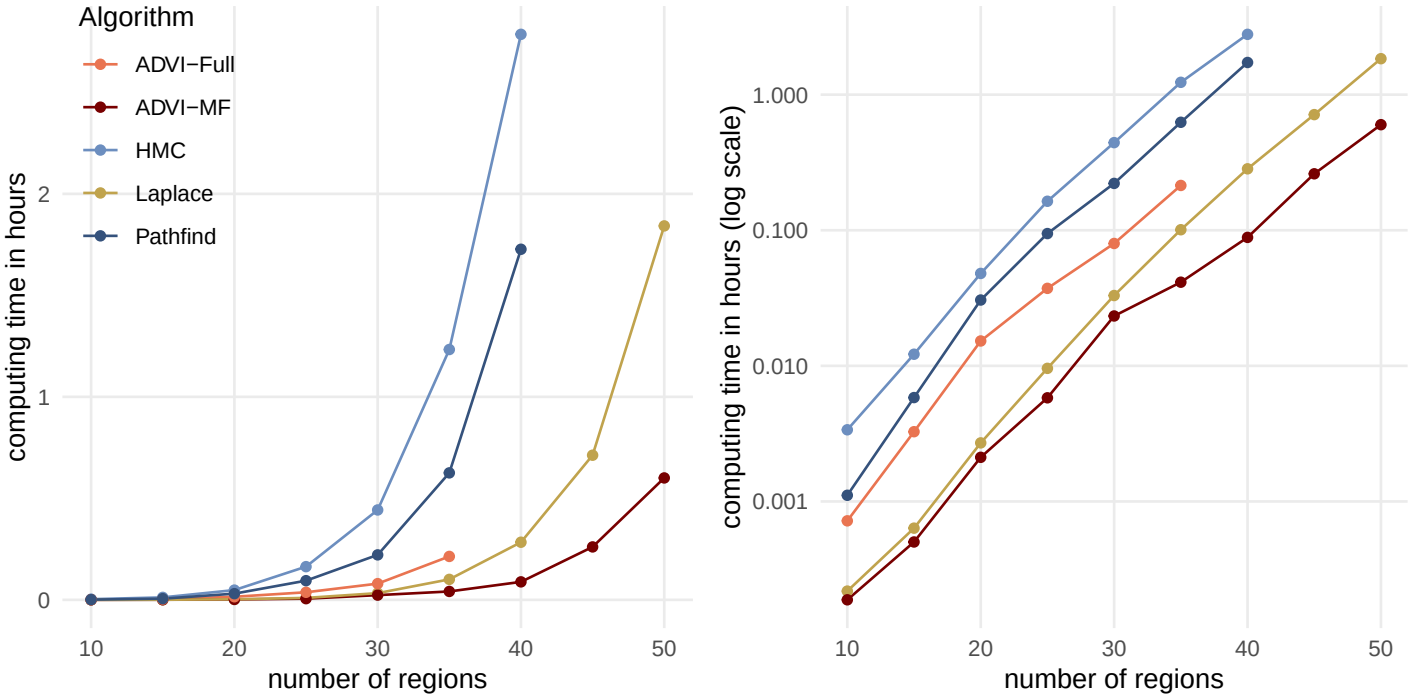
Variational inference - Poisson regression in Turing.jl

```
# Variational inference for posterior of β
τ = 10    # Prior standard deviation
model = poissonReg(y, X, τ)
nSamples = 10
nGradSteps = 1000
approx_post = vi(model, ADVI(nSamples, nGradSteps))
βsample = rand(approx_post, 1000)
```

Variational inference - Poisson regression in Turing.jl



Analyzing brain connectivity from fRMI data



Some resources for further study

- There are many good Bayesian textbooks, for example:
  - Gelman et al (2013). Bayesian Data Analysis
  - Bishop (2006). Pattern Recognition and Machine Learning
  - McElreath (2022). Statistical Rethinking.
  - Bernardo and Smith (1994). Bayesian Theory.
- Here are some of my own materials:
  - Bayesian Learning - a gentle introduction. Book in progress.
  - Bayesian Learning course - slides, computer labs and exams.
  - Advanced Bayesian Learning course - slides and computer labs.
  - Bayesian Learning - Observable Javascript widgets.
- Turing tutorials with neural nets and variational inference.
- The excellent Stan user guide has a lot of examples.
- PyMC is one of the many PPL for Bayes in Python.