

# Example-04: Stochastic transport model

---

```
1 # Uwe Hernandez Acosta, Mikhail Mikhasenko, 03.06.2026
```

```
1 begin
2   using Distributions
3   using Plots
4 end
```

```
1 theme(:boxed)
```

## Parameters of distribution

---

they are fixed, or parametric

$\sigma = 0.2$

```
1  $\sigma = 0.2$ 
```

gun (generic function with 1 method)

```
1 gun(; pars) = Normal(pars. $\mu$ , $\sigma$ )
```

## Transport code

---

is numerical, on cluster, heavy

transport (generic function with 1 method)

```
1 function transport(x; pars)
2   (; a, b) = pars
3   # should have stochastic component
4    $\Delta\alpha = (2\text{rand}()+1) * a$ 
5    $(x+\Delta\alpha)^3 * \exp(x-b)$ 
6 end
```

## Postprocessing

---

projecting data, could be heavy

## Full chain

---

```
x ~ gun(; pars)
y ~ transport(x; pars)
z ~ post(y)
```

```
post (generic function with 1 method)
```

```
post(x) = x/2
```

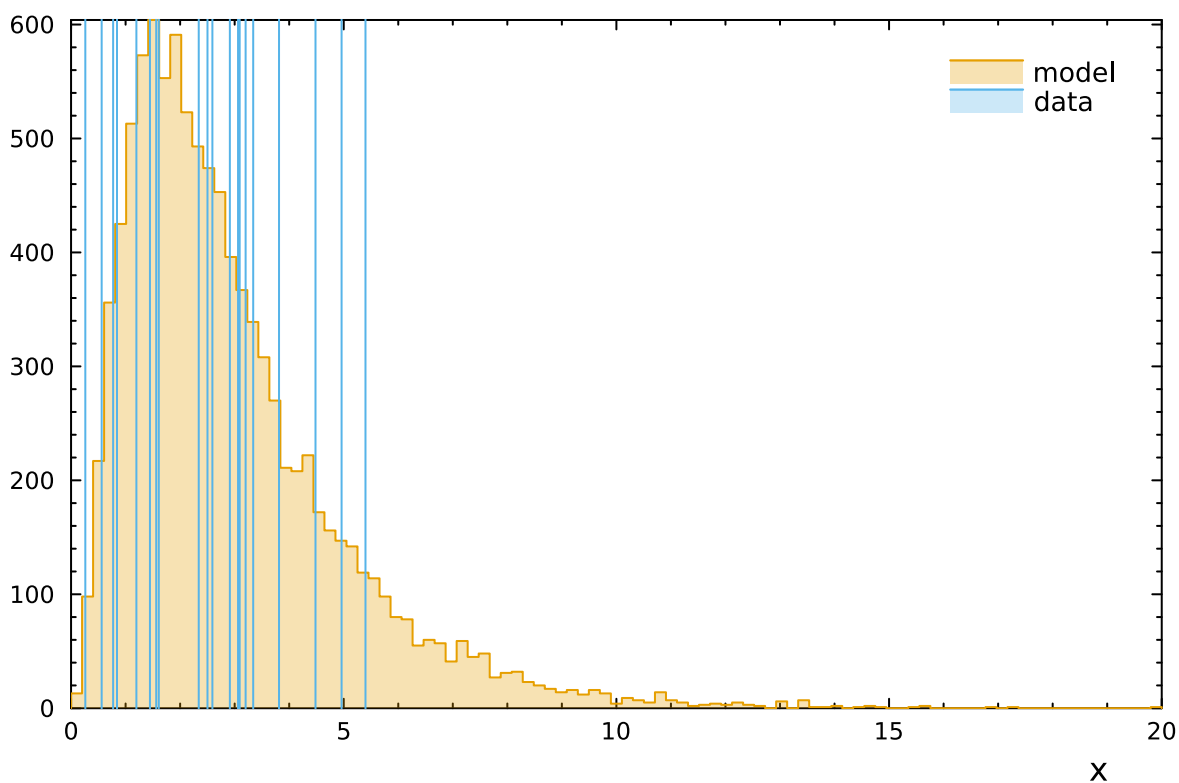
```
analysis (generic function with 1 method)
```

```
function analysis(pars)
  #
  gun_dist = gun(; pars)
  x = rand(gun_dist, 10_000)
  #
  y = transport.(x; pars)
  #
  z = post.(y)
end
```

```
data = [3.81359, 2.91195, 3.20085, 3.09185, 3.34005, 4.96067, 0.842412, 2.34128,
3.06224, 2.59162, 2.5017, 5.39892, 1.19806, 1.60855, 1.44647, 0.771489, 0.26153,
1.56184, 0.561171, 4.4823];
```

```
glob_pars = (a = 0.1, b = 0.3,  $\mu$  = 1.1)
```

```
glob_pars = (a = 0.1, b = 0.3,  $\mu$  = 1.1)
```



```
begin
  bins = range(0,20,100)
  stephist(analysis(glob_pars); bins, xlim=(0, 20), fill=0, fillalpha=0.3, c=2,
  lab="model", xlabel="x")
  vline!(data, xlim=(0, 20), fill=0, fillalpha=0.3, c=3, lab="data")
end
```

# Likelihood

---

$z \sim \text{post}(y)$

determines the model distribution

The likelihood is computed by comparing the given data distribution with the simulation model e.g. by

- binning the model pdf, and evaluating data
- KL divergence

## Missing execution model

---

- $\sim$  is incomplete: drawing algorithm, seed, number of events [there is no way to enforce the standard].
- Execution model: GPU vs CPU
- Where the type of  $x$ ?

```
functions: {[
  {
    name: "transport",
    input: x::Float64,
    output: y::Float64
  }]
}
```

### Why execution is needed?

clearly, we want to run the analysis – this is the job of the engine.

### Why naive execution model is not working?

- Does not scale.
- Inconsistent between different languages.
- Not reproducible due to lack of technical depth.

### More sophisticated execution model

fixes the technical details IN FORMAT, in a modular way.