Bayesian Machine Learning for Scientific Research

Maximizing information from data

Ezequiel Alvarez
sequi@unsam.edu.ar
ICTP-SAIFR October 2024

Previous lectures

Lecture 1

Bayesian: assume data being sampled from a PDF, infer its parameters and learn the internal structure of the data

To learn the PDF of the data and then being able to assess, predict , generate, etc.

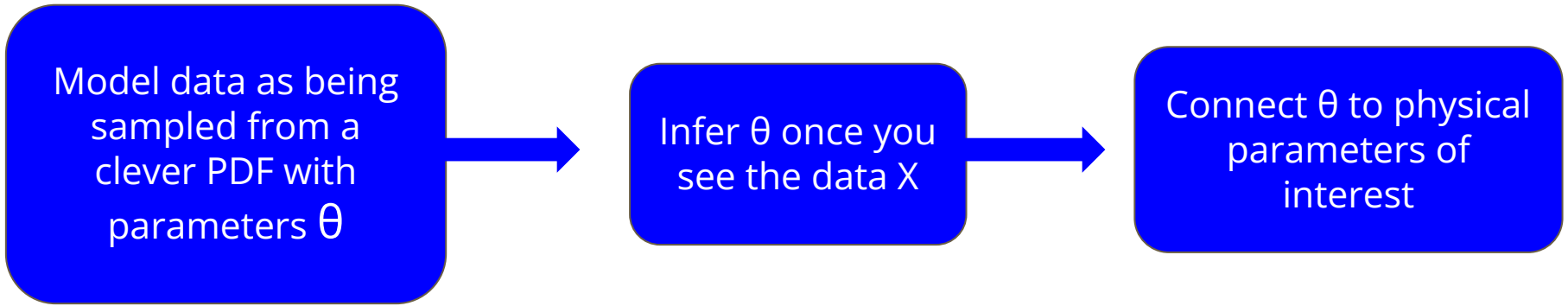
More scientific

Lecture 1

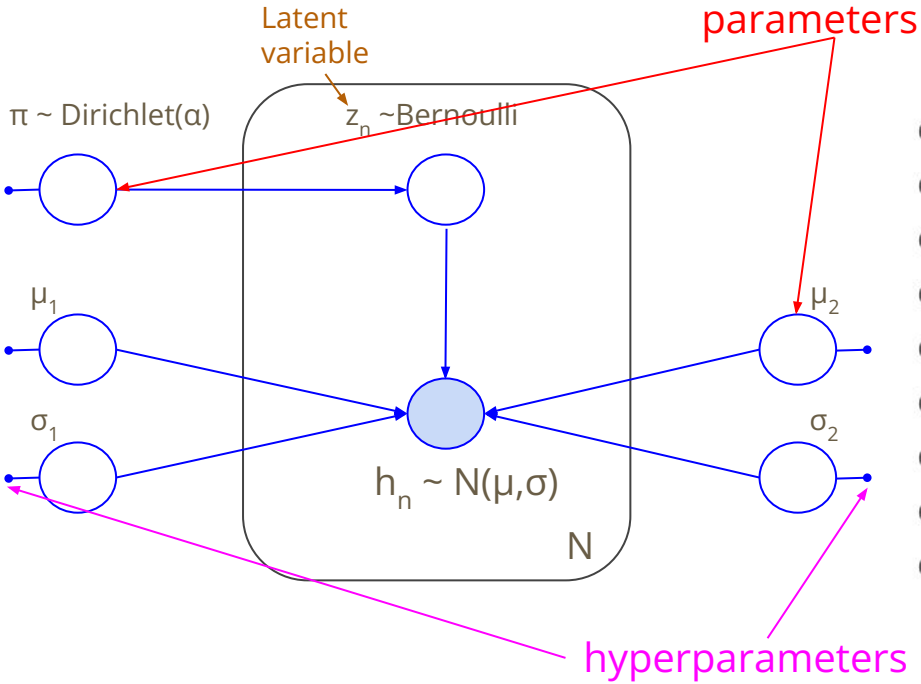
Bayes Theorem:

$$p(\theta|X) = \frac{p(X|\theta) * p(\theta)}{p(x)}$$

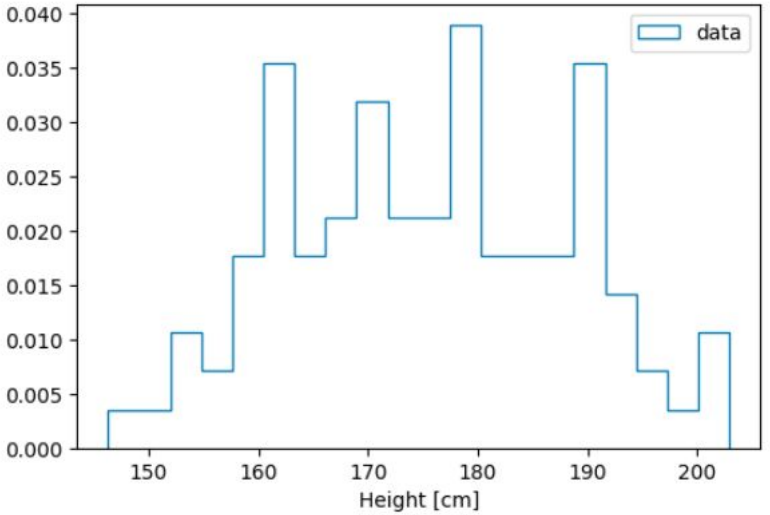
Our utility: X = data, θ=parameters



Lecture 1

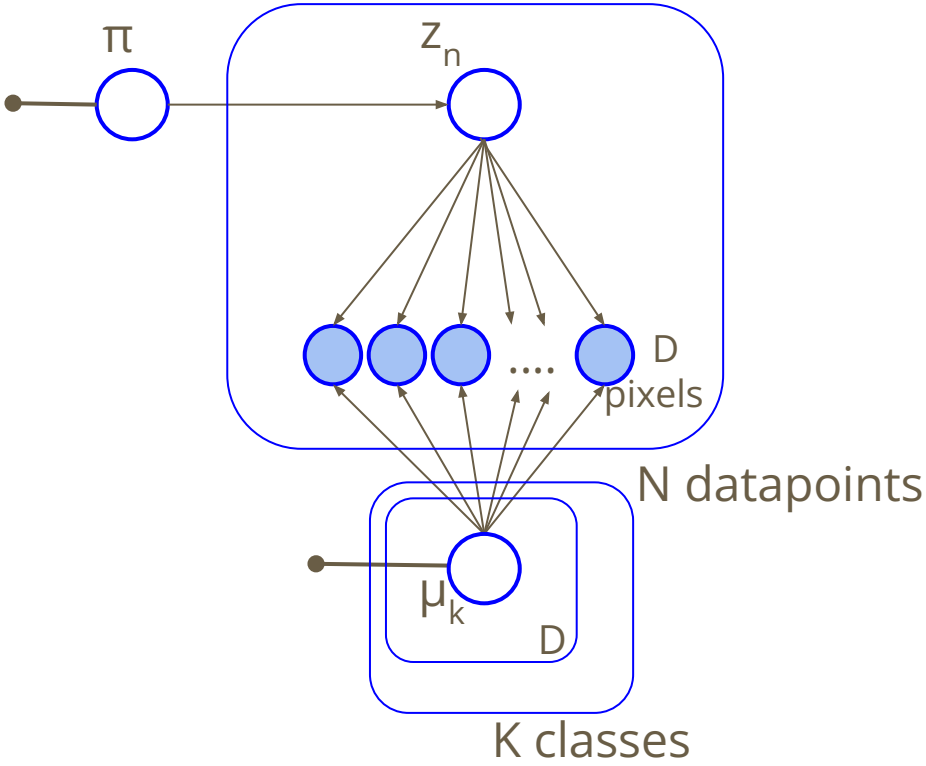


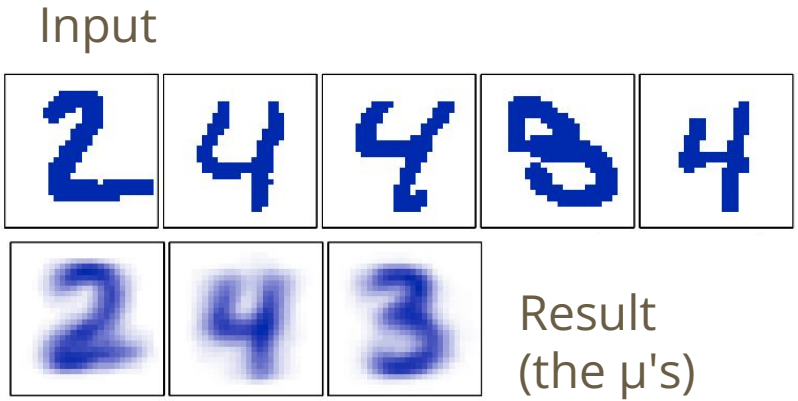
Graphical Model



1D Gaussian Mixture

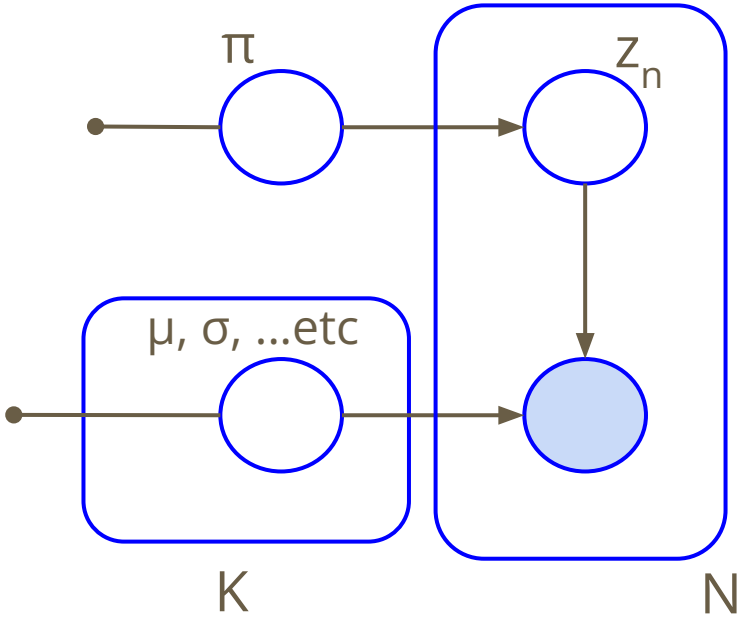
Lecture 2: Mixture of Bernoulli





Scientifically: z_n is the probability of each class

Lecture 3: Mixture Models



$$p(x_n|\Theta) = \sum_{k=1}^{K} \pi_k \, p(x_n|\theta_k)$$

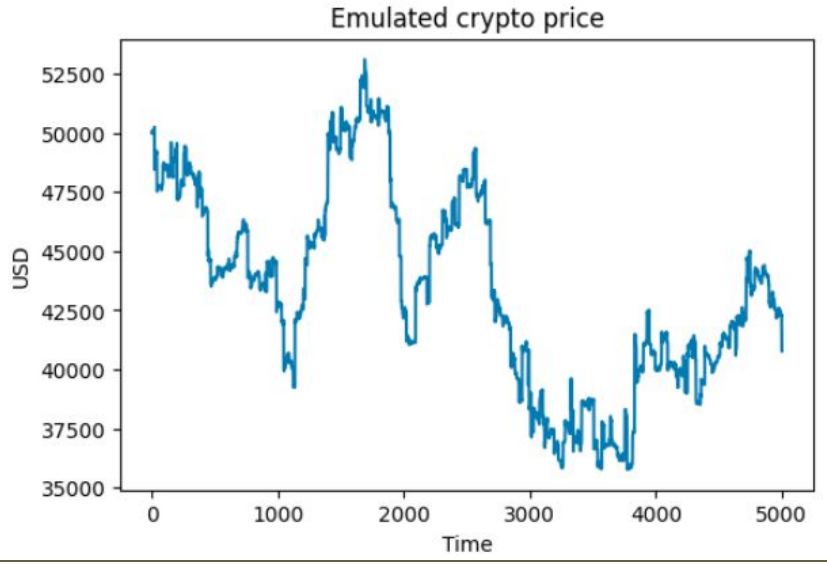
$$\log p(X|\Theta) = \sum_{n=1}^{N} \log\Big(\sum_{k=1}^{K} \pi_k \, p(x|\theta_k) \Big)$$

Mixture Models: Crypto price ?

Real Bitcoin price Aug to Sep

```
sigmas = [10,100,1000]
crypto_price = [50000]
np.random.seed(1)
z = np.random.choice([0,1,2], p=[0.7,0.25,0.05], size=5000)
for zn in z:
    crypto_price.append(cripto_price[-1]+np.random.normal(0,sigmas[zn]))
```

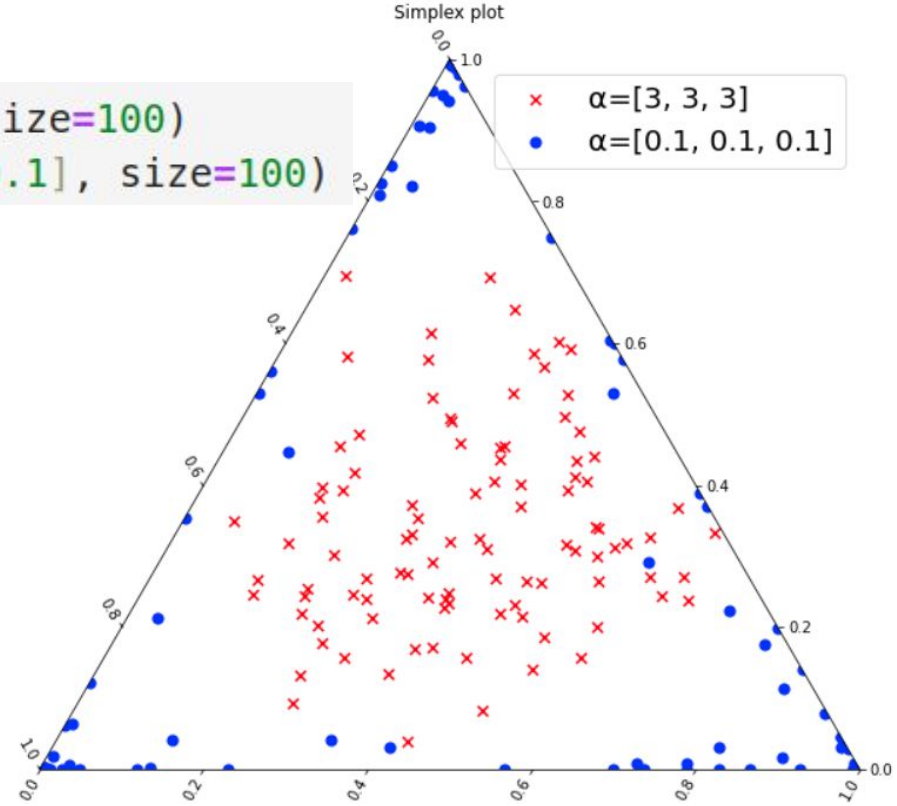
```
plt.plot(cripto_price)
plt.title('Emulated crypto price')
plt.ylabel('USD')
plt.xlabel('Time')
plt.show()
```



Lecture 3: Dirichlet Distribution

```
sample1 = np.random.dirichlet([3,3,3], size=100)
sample2 = np.random.dirichlet([0.1,0.1,0.1], size=100)
```

p(X) = *Dir*(X; α) = $$\frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$



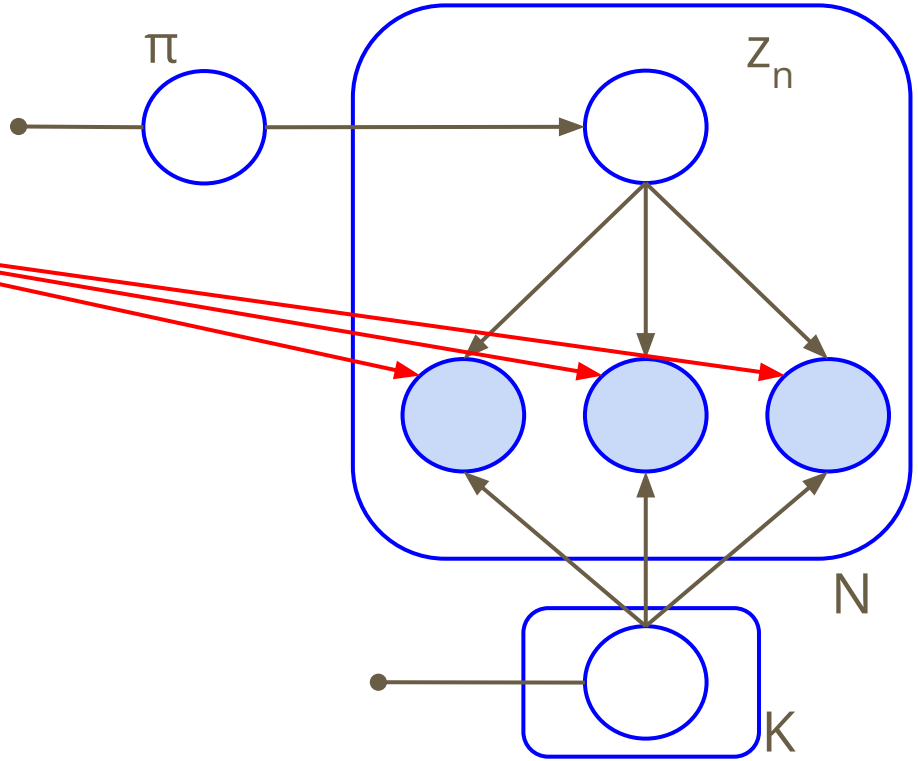
Lecture 3: Mixture Models, where is the hack ?

Conditionally independence

Once the class is defined,
they are independent

Huge reduction in number of parameters to infer!

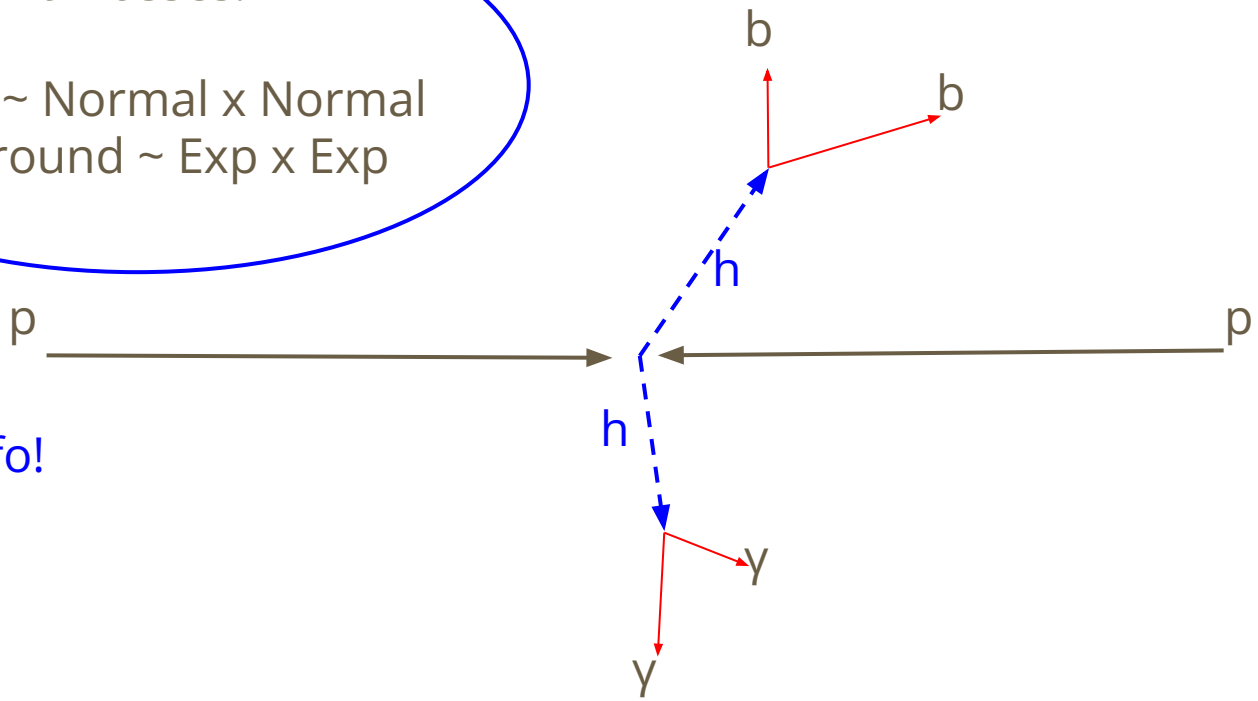
better grip!
(B^D vs B*D)



Lecture 3: 2D Mixture Model, hh → bbγγ

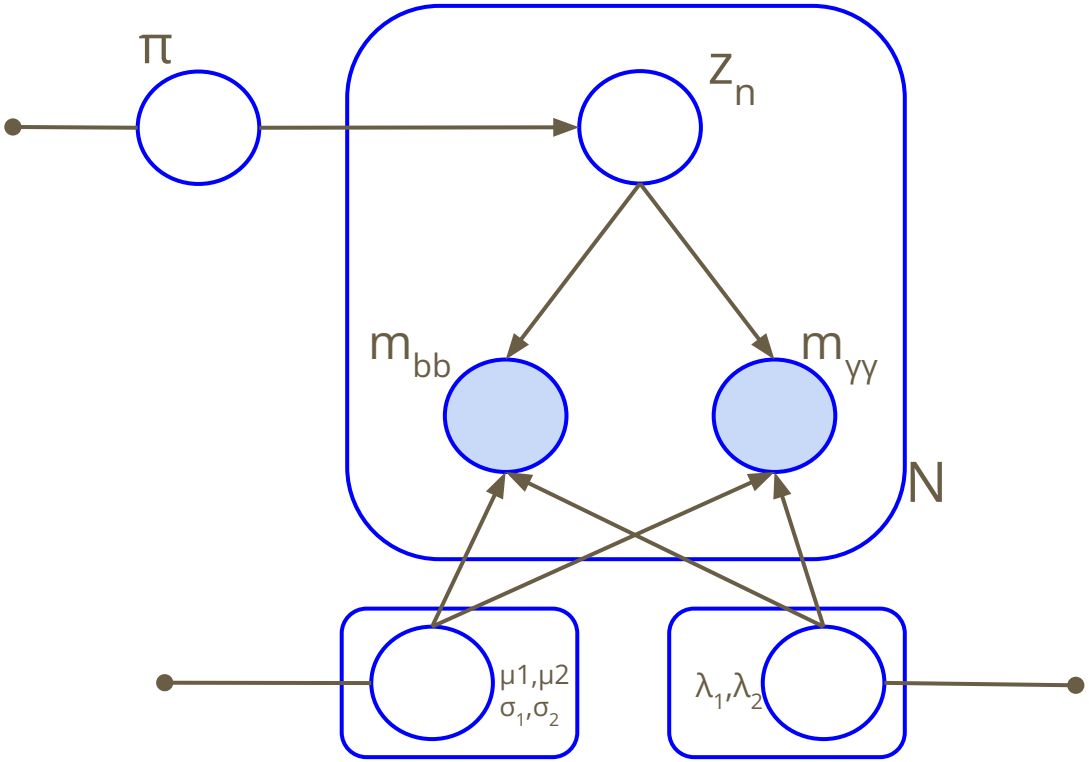
Invariant masses:

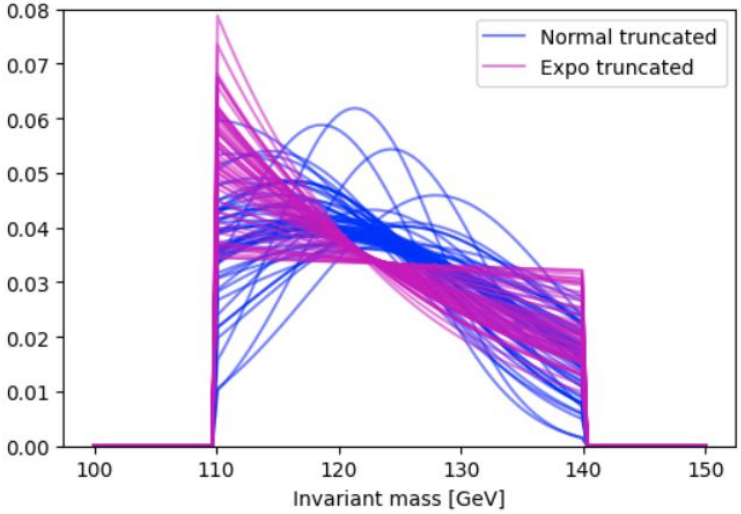
Signal ~ Normal x Normal
Background ~ Exp x Exp



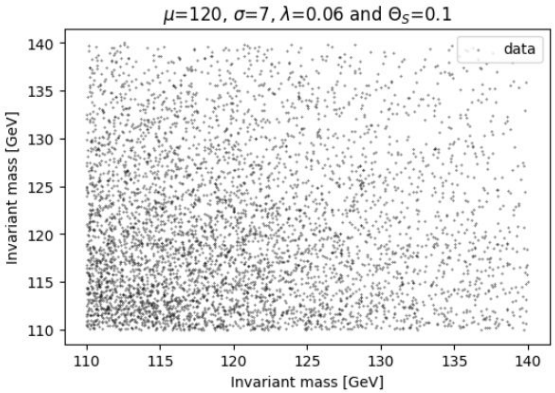
Prior info!

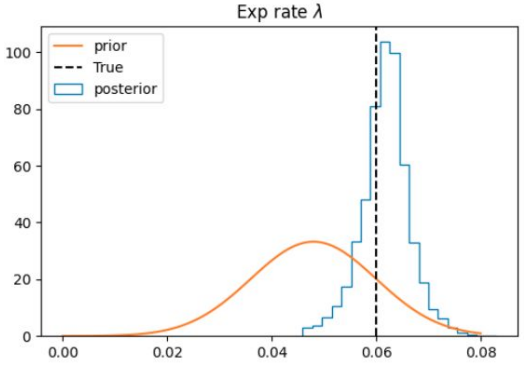
Lecture 3: 2D Mixture Model, hh → bbγγ

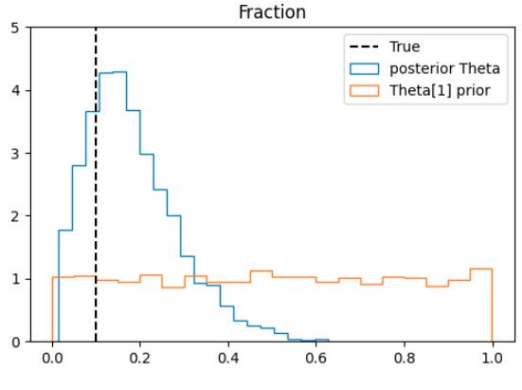


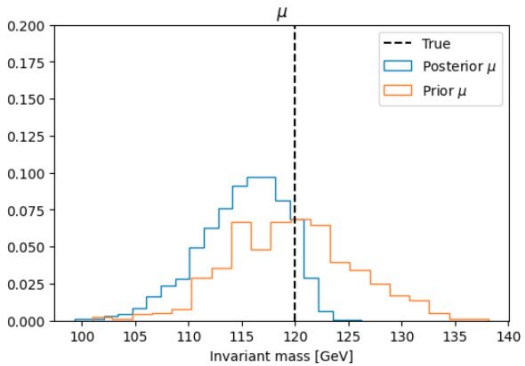


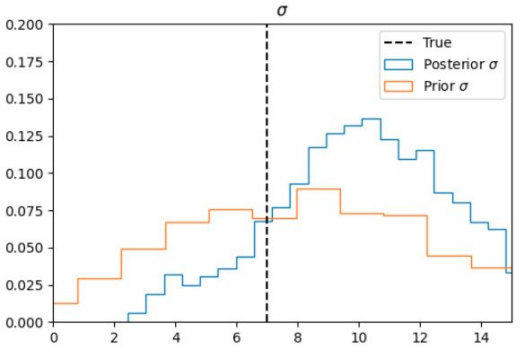
Lecture 3: The impossible…. @10%



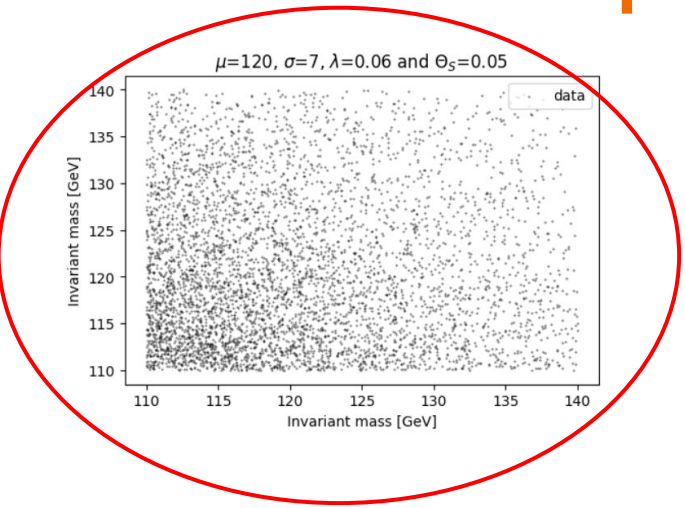


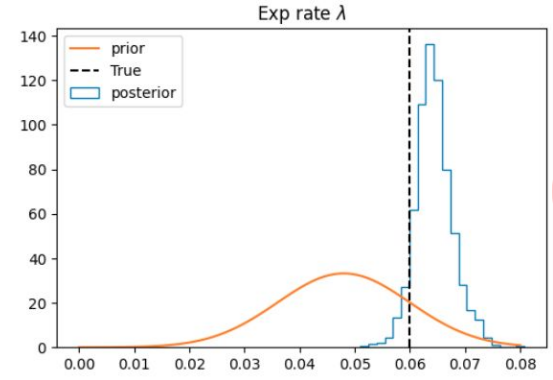


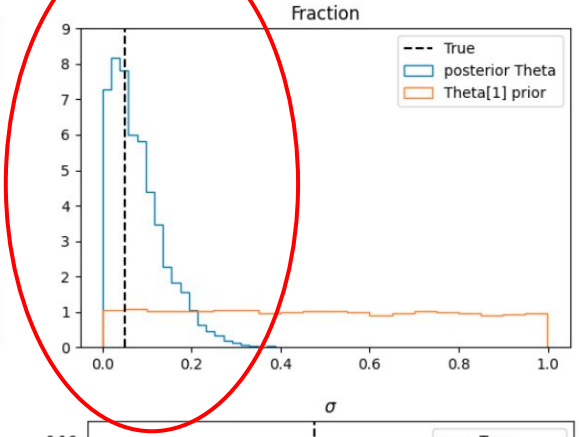


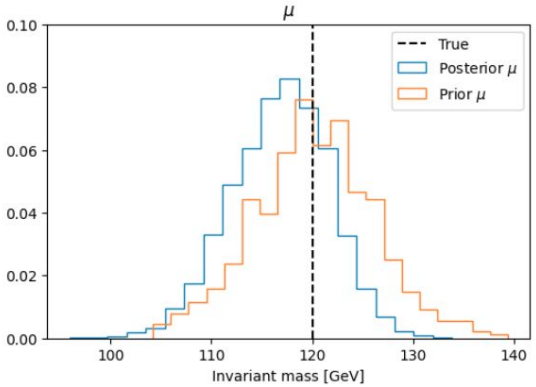


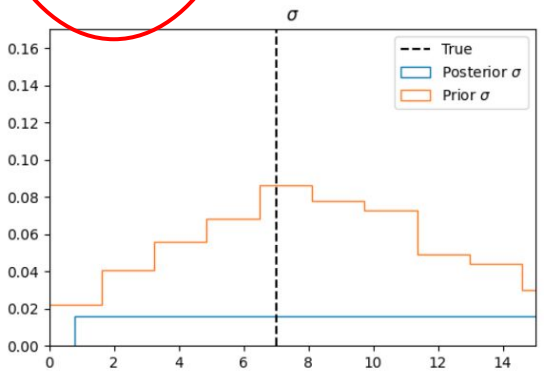
Lecture 3: The impossible.... @5%!









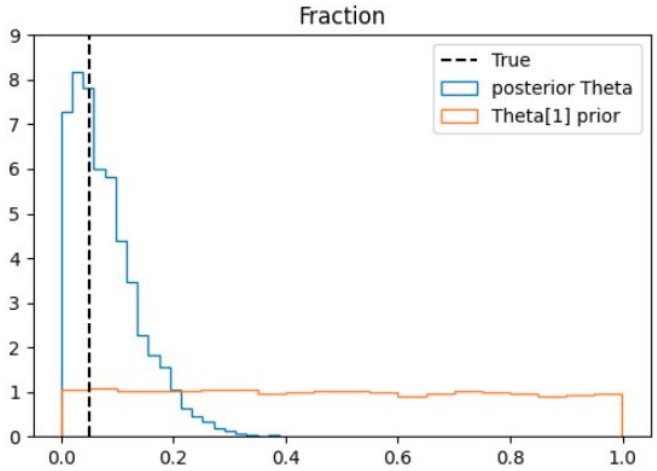


Correlation
Correlation
Correlation!

Assessment in Bayesian ML

Lecture 4

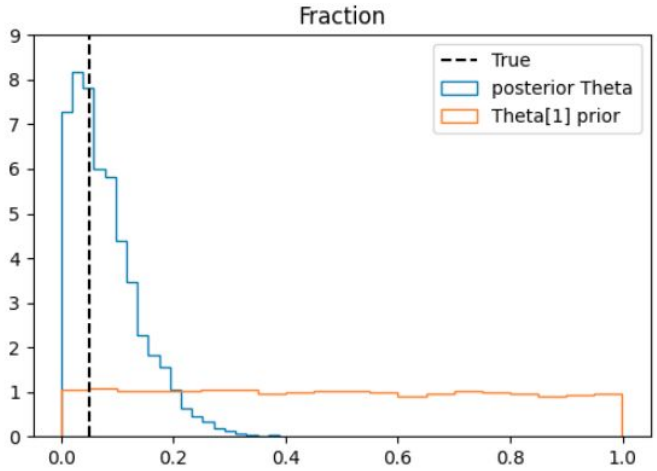
Assessment in Bayesian ML



Once you have a result from real data, you want to test that...

- The sampling is unbiased
- The model works correctly
- The model is (fairly) correct for the data

Assessment in Bayesian ML



Once you have a result from real data, you want to test that...

- The sampling is unbiased

Sampling diagnostics

- The model works correctly

Fake data to test it

- The model is (fairly) correct for the data

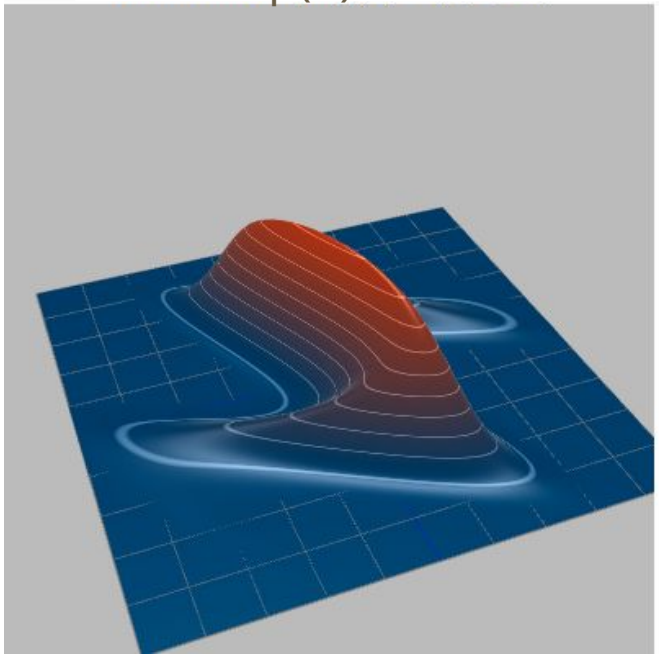
Posterior Predictive Check

Sampling Assessment

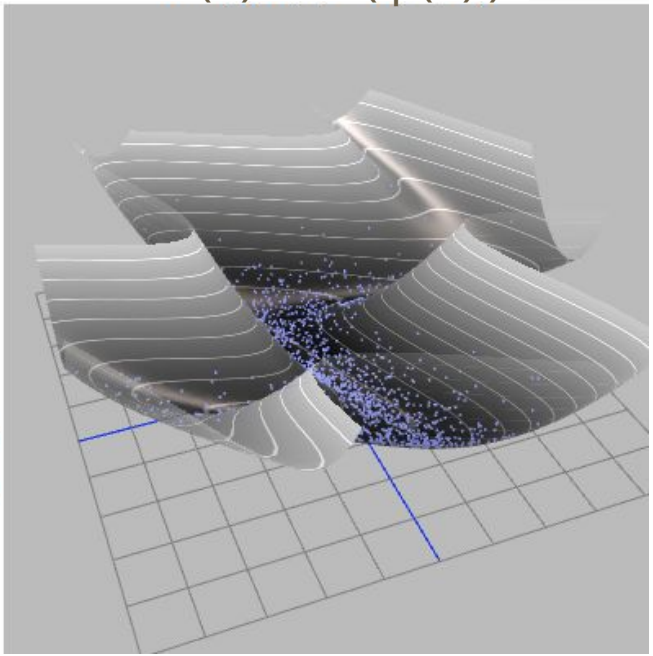
@Hamiltonian MonteCarlo

MCMC Sampling with Hamiltonian MC

$p(\theta)$



$V(\theta) = -\ln(p(\theta))$

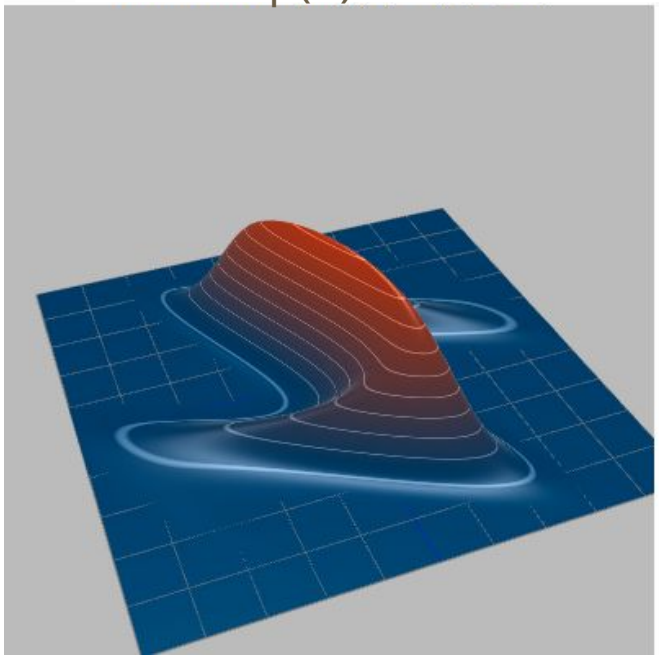


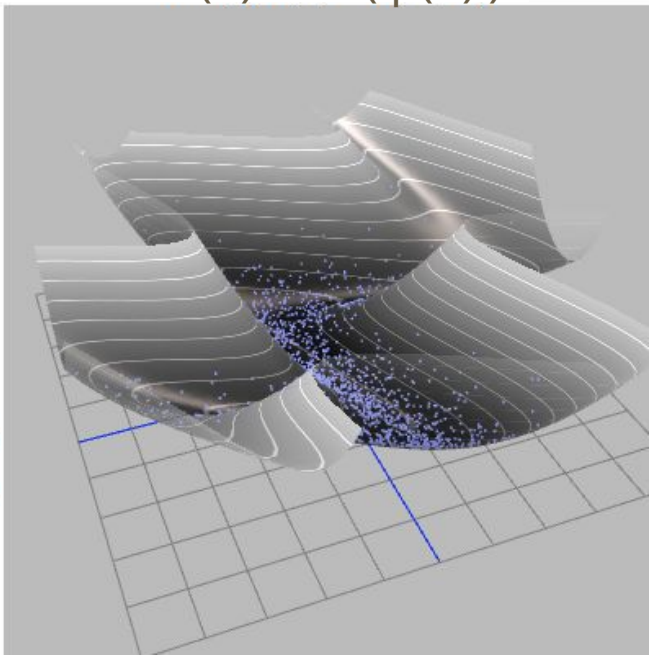
MCMC Sampling with Hamiltonian MC

$p(\theta)$

$V(\theta) = -\ln(p(\theta))$

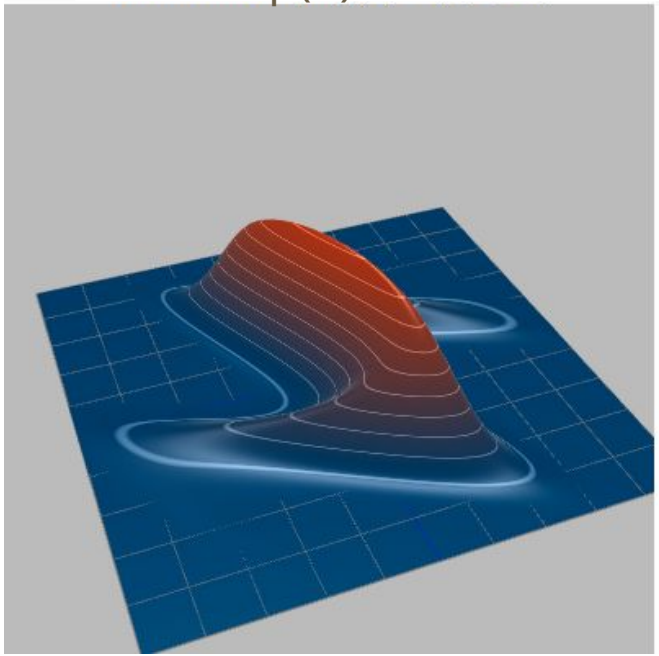
We know all its gradients



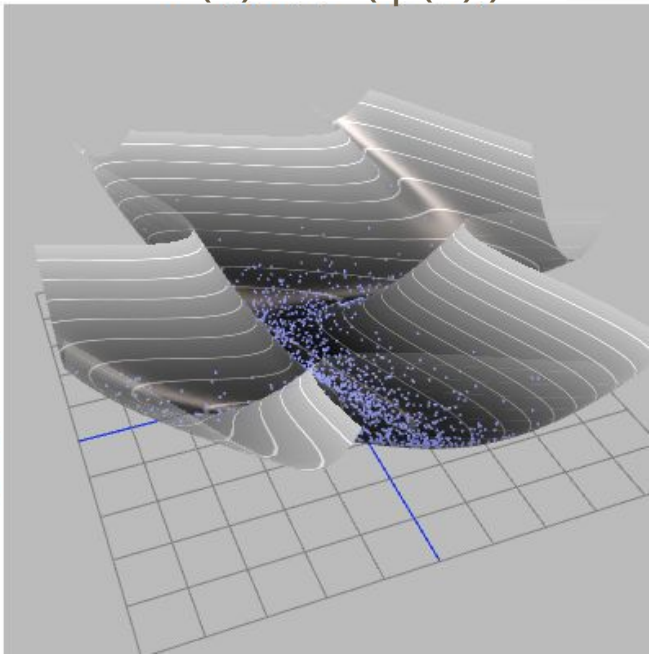


MCMC Sampling with Hamiltonian MC

$p(\theta)$



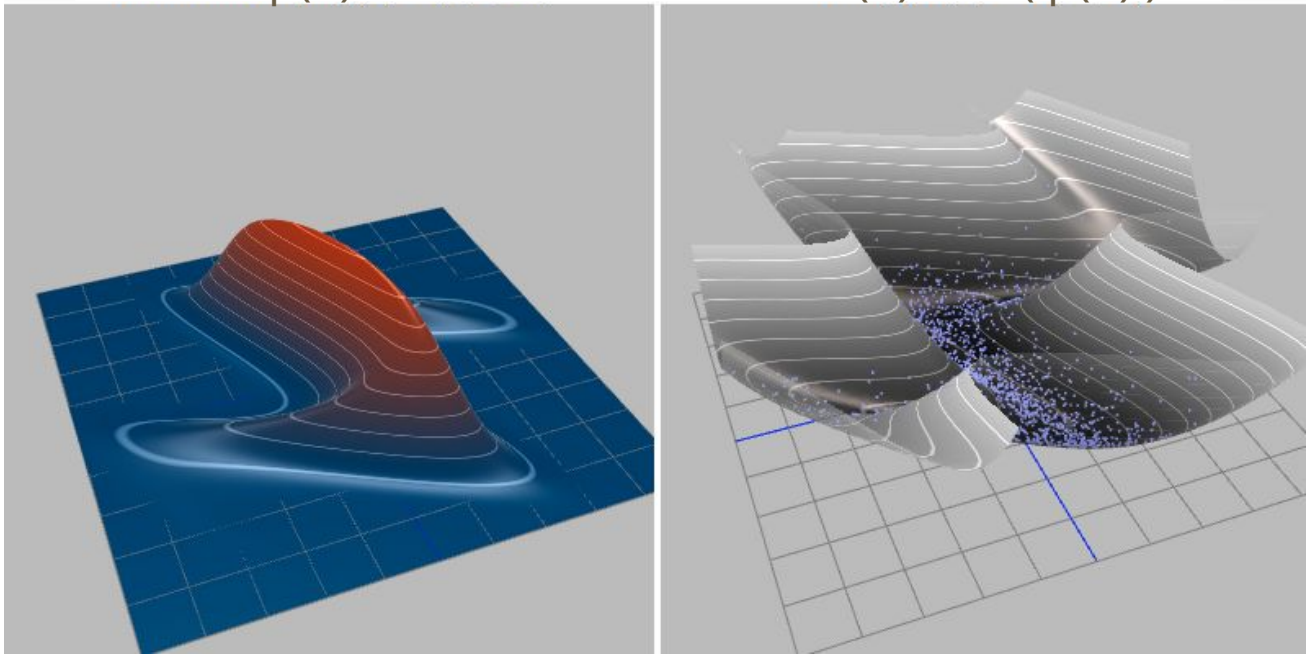
$V(\theta) = -\ln(p(\theta))$



Simulate trajectory of fictitious particle

MCMC Sampling with Hamiltonian MC

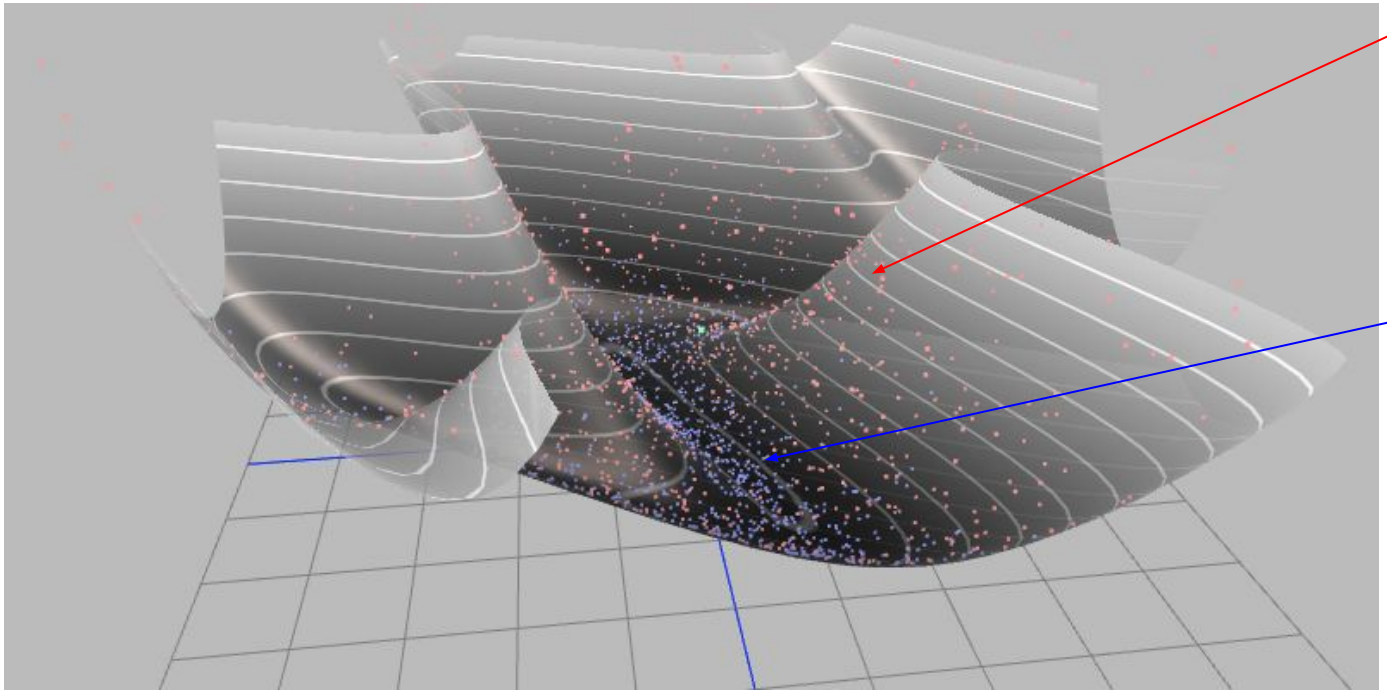
$p(\theta)$ $V(\theta) = -\ln(p(\theta))$



Simulate trajectory of fictitious particle

Introduce stochastic energy loss in order to stay more time in the larger probability regions

MCMC Sampling with Hamiltonian MC



Numerical integration yields some divergent trajectories (rejected)

Accepted samples

MCMC Sampling with Hamiltonian MC

parameters draws	lp__	accept_stat__	stepsize__	treedepth__	n_leapfrog__	divergent__	energy__	mu	sigma
0	-32598.734453	0.994393	0.176560	3.0	11.0	0.0	32600.239461	120.773958	6.188426
1	-32601.510781	0.808358	0.149106	4.0	15.0	0.0	32604.731098	115.727014	10.256436
2	-32597.717577	0.973526	0.137055	3.0	7.0	0.0	32599.468725	120.248254	6.372511
3	-32597.238541	0.998011	0.141012	4.0	27.0	0.0	32599.385174	119.383923	7.435660
4	-32598.284739	0.988675	0.176560	4.0	15.0	0.0	32600.078479	119.809050	6.475248

p(sample) _____HMC numbers_____/ parameters

MCMC Sampling with Hamiltonian MC

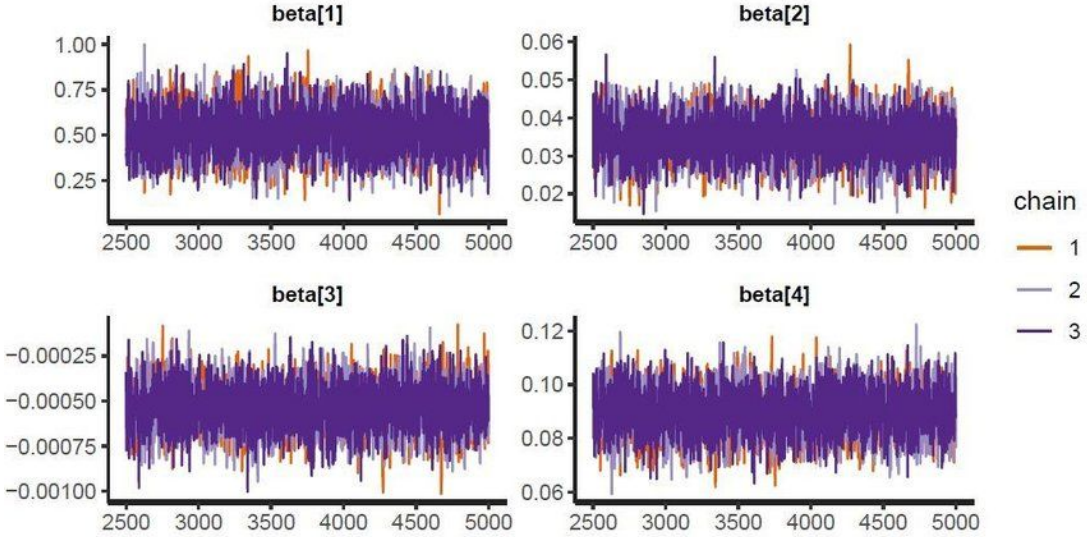
parameters draws	lp__	accept_stat__	stepsize__	treedepth__	n_leapfrog__	divergent__	energy__	mu	sigma
0	-32598.734453	0.994393	0.176560	3.0	11.0	0.0	32600.239461	120.773958	6.188426
1	-32601.510781	0.808358	0.149106	4.0	15.0	0.0	32604.731098	115.727014	10.256436
2	-32597.717577	0.973526	0.137055	3.0	7.0	0.0	32599.468725	120.248254	6.372511
3	-32597.238541	0.998011	0.141012	4.0	27.0	0.0	32599.385174	119.383923	7.435660
4	-32598.284739	0.988675	0.176560	4.0	15.0	0.0	32600.078479	119.809050	6.475248

p(sample) _____HMC numbers_____/ parameters

Simple solutions, e.g. make a more detailed exploration

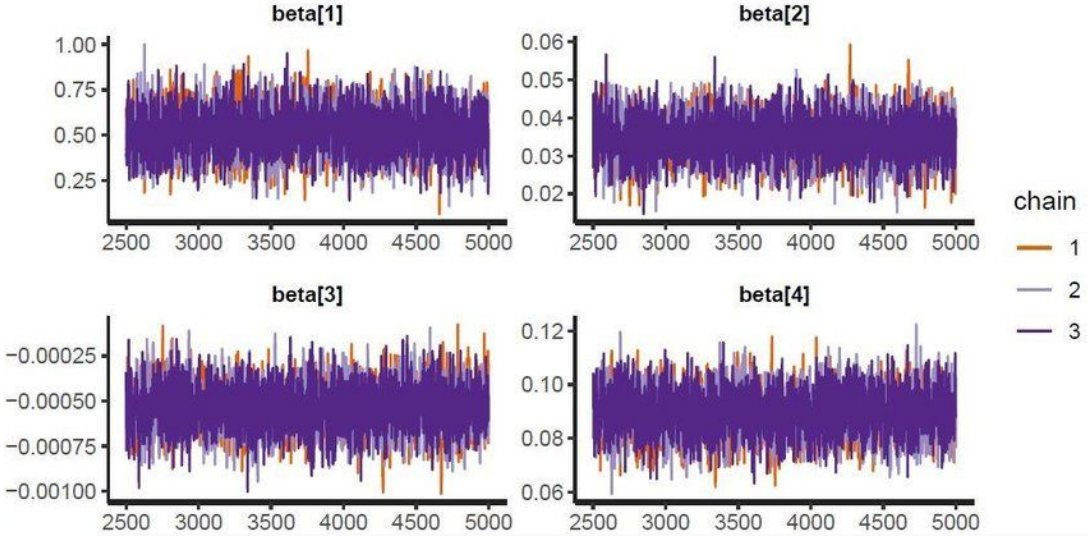
```
fit = model.sample(num_chains=4, num_samples=1000, delta=0.95)
```

MCMC Sampling with Hamiltonian MC



Check that all (independent) chains end up exploring approximately the same!

MCMC Sampling with Hamiltonian MC



Check that all (independent) chains end up exploring approximately the same!

R-hat < 1.05

```
R-hat = sqrt((Variance Between Chains) / (Variance Within Chains))
```

MCMC Sampling with Hamiltonian MC

```
azdata = az.from_pystan(fit)
summary = az.summary(azdata)
summary
```

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
mu	118.861	1.320	116.425	120.933	0.055	0.039	691.0	570.0	1.01
sigma	7.770	1.120	5.868	9.878	0.046	0.033	660.0	702.0	1.01
lambda0	0.062	0.004	0.054	0.069	0.000	0.000	1114.0	842.0	1.00
theta[0]	0.348	0.089	0.200	0.505	0.004	0.003	630.0	678.0	1.01
theta[1]	0.652	0.089	0.495	0.800	0.004	0.003	630.0	678.0	1.01

Check that all (independent) chains end up exploring approximately the same!

R-hat < 1.05

```
R-hat = sqrt((Variance Between Chains) / (Variance Within Chains))
```

MCMC Sampling with Hamiltonian MC

```
azdata = az.from_pystan(fit)
summary = az.summary(azdata)
summary
```

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
mu	118.861	1.320	116.425	120.933	0.055	0.039	691.0	570.0	1.01
sigma	7.770	1.120	5.868	9.878	0.046	0.033	660.0	702.0	1.01
lambda0	0.062	0.004	0.054	0.069	0.000	0.000	1114.0	842.0	1.00
theta[0]	0.348	0.089	0.200	0.505	0.004	0.003	630.0	678.0	1.01
theta[1]	0.652	0.089	0.495	0.800	0.004	0.003	630.0	678.0	1.01

Check that all (independent) chains end up exploring approximately the same!

R-hat < 1.05

```
R-hat = sqrt((Variance Between Chains) / (Variance Within Chains))
```

There are a few more diagnostic indicators, but this plus some adjustments in adapt, max_tree, etc. is usually enough

Model works correctly?

Model works correctly ?

Never use a model before testing it

Model works correctly ?

Never use a model before testing it

- Create fake data as you expect it to be the real data

Model works correctly ?

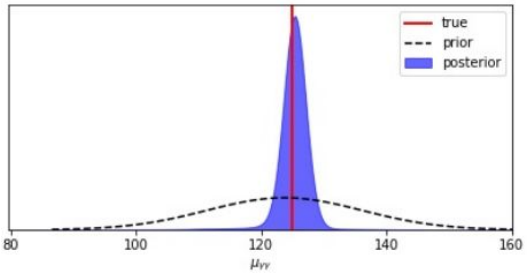
Never use a model before testing it

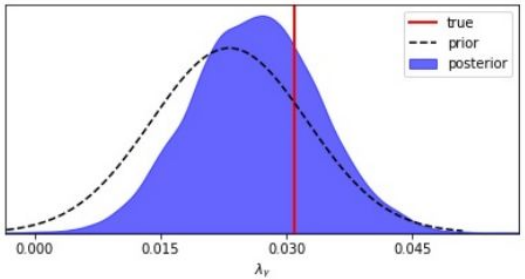
- Create fake data as you expect it to be the real data
- You know the trues, start then with biased priors

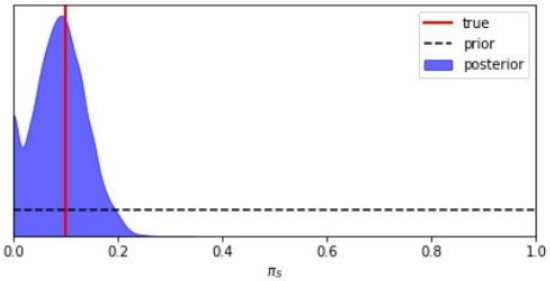
Model works correctly ?

Never use a model before testing it

- Create fake data as you expect it to be the real data
- You know the trues, start then with biased priors
- Verify that the inference process approaches the posterior to the (known) trues





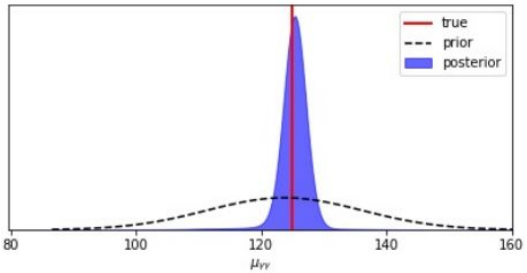


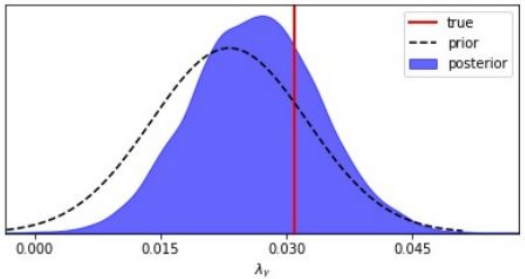
Model works correctly ?

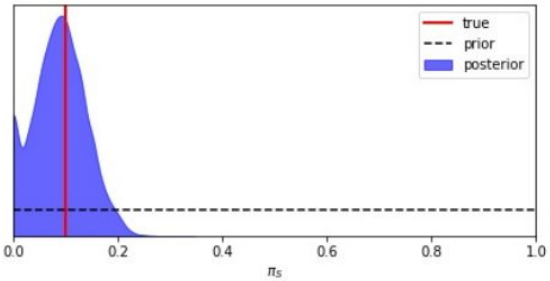
Never use a model before testing it

- Create fake data as you expect it to be the real data
- You know the trues, start then with biased priors
- Verify that the inference pro
  trues

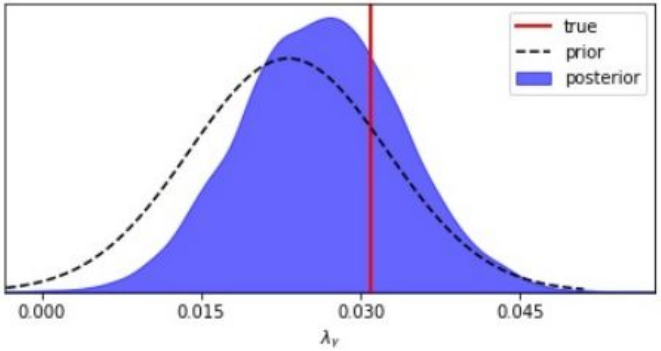
There is no general rule on what is satisfactory, each problem has its own expectations!







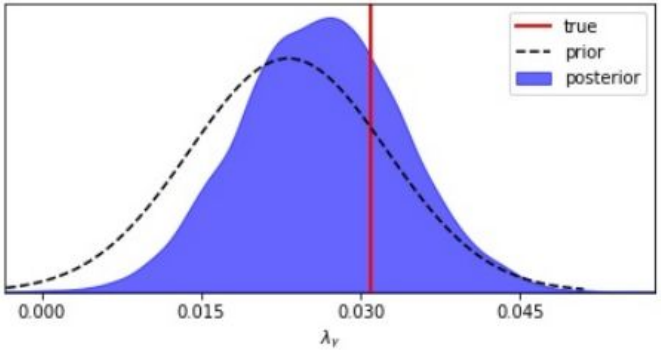
Model works correctly ?



You have (at least) 2 metrics to test

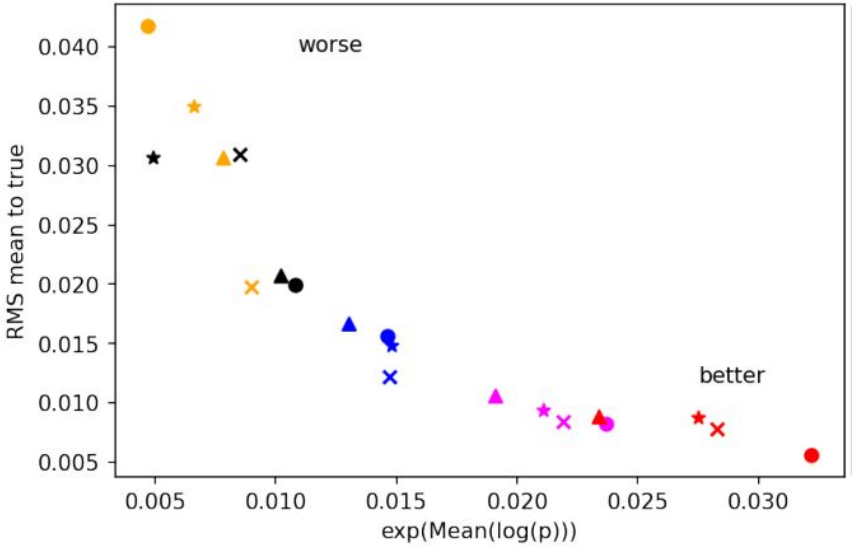
- Density at the *true*
- Distance *mean* to *true*

Model works correctly ?



You have (at least) 2 metrics to test

- Density at the *true*
- Distance *mean* to *true*



Posterior Predictive Check

The probability of the data

How good is the model to explain the data

Posterior predictive check

How good is the model to explain the data ?

Posterior predictive check

How good is the model to explain the data ? Once you infer:

What is the probability
of the data within the model ?

Posterior predictive check

How good is the model to explain the data ? Once you infer:

What is the probability
of the data within the model ?

How we measure what is
good and what is bad ??

What is the probability of the data ?

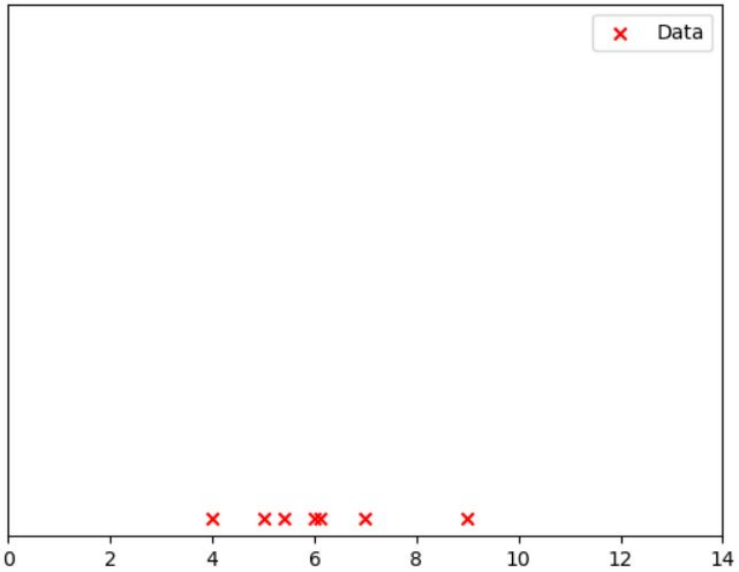
- Data is usually multidimensional

What is the probability of the data ?

- Data is usually multidimensional
- After inference we can easily sample tons of synthetic data!

What is the probability of the data ?

- Data is usually multidimensional
- After inference we can easily sample tons of synthetic data!

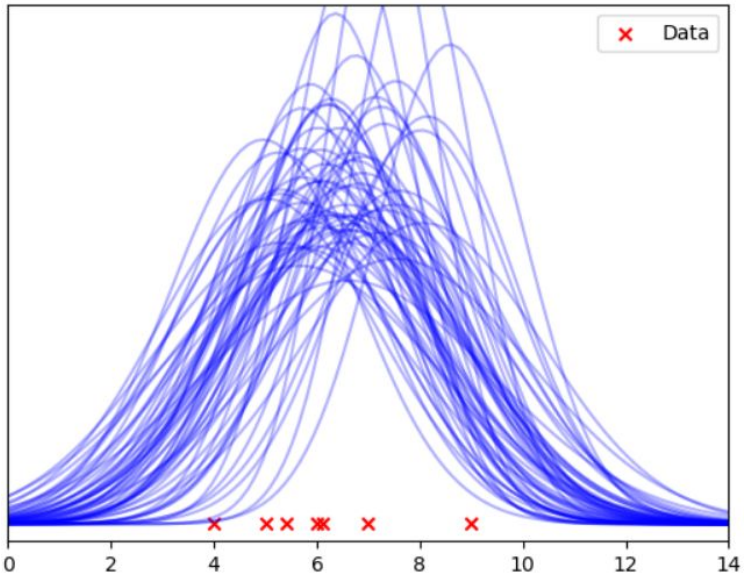


1D case

You have some data

What is the probability of the data ?

- Data is usually multidimensional
- After inference we can easily sample tons of synthetic data!



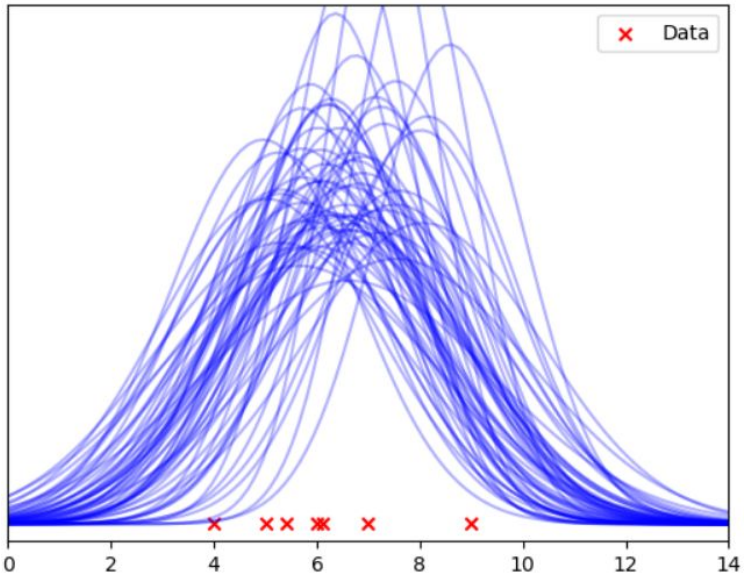
1D case

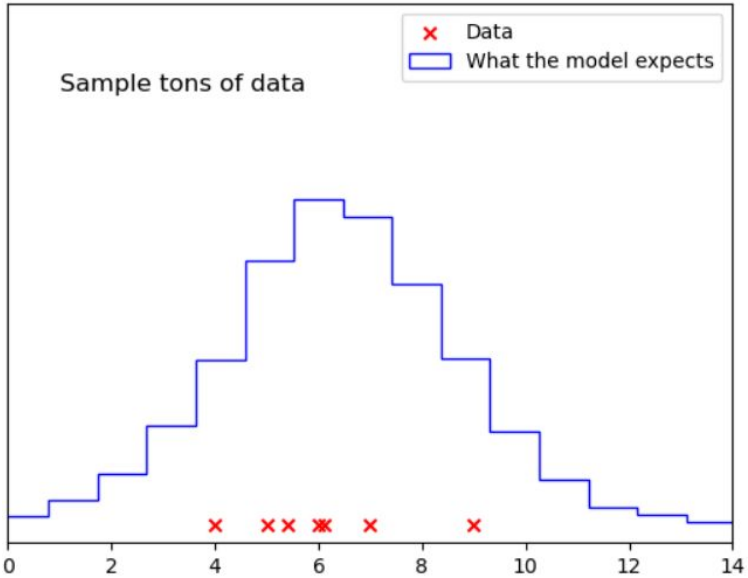
You have some data

Solve the inference problem with your model

What is the probability of the data ?

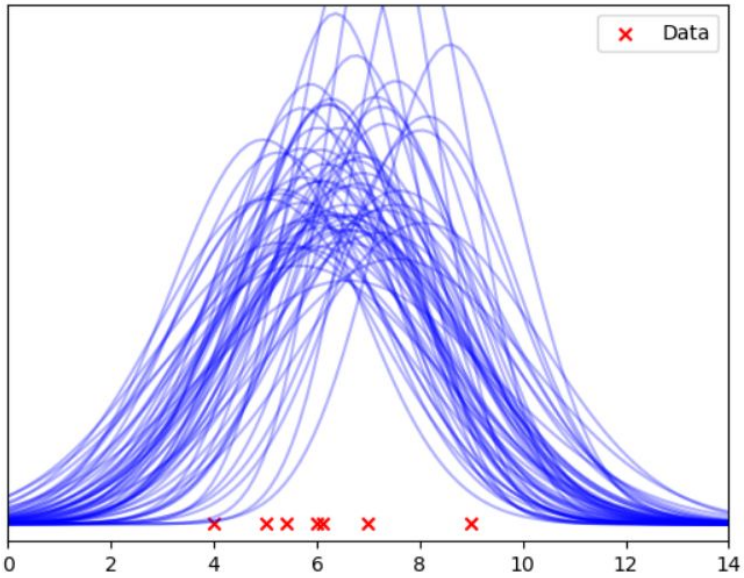
- Data is usually multidimensional
- After inference we can easily sample tons of synthetic data!

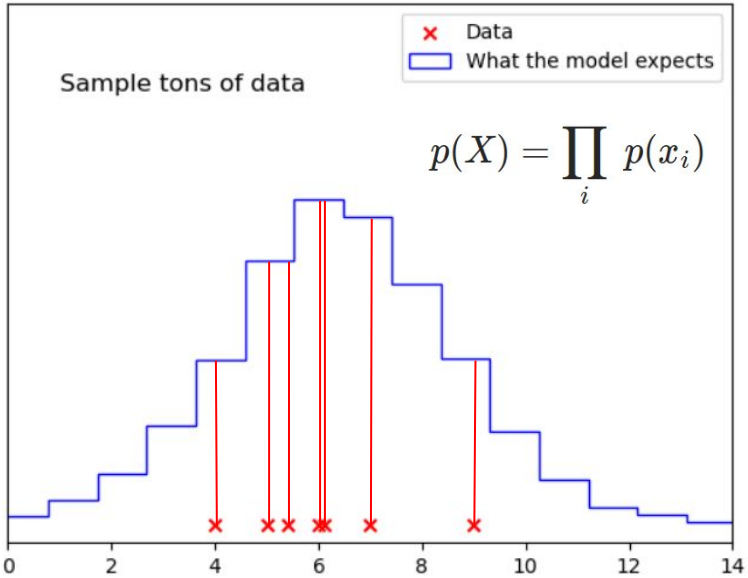




What is the probability of the data ?

- Data is usually multidimensional
- After inference we can easily sample tons of synthetic data!

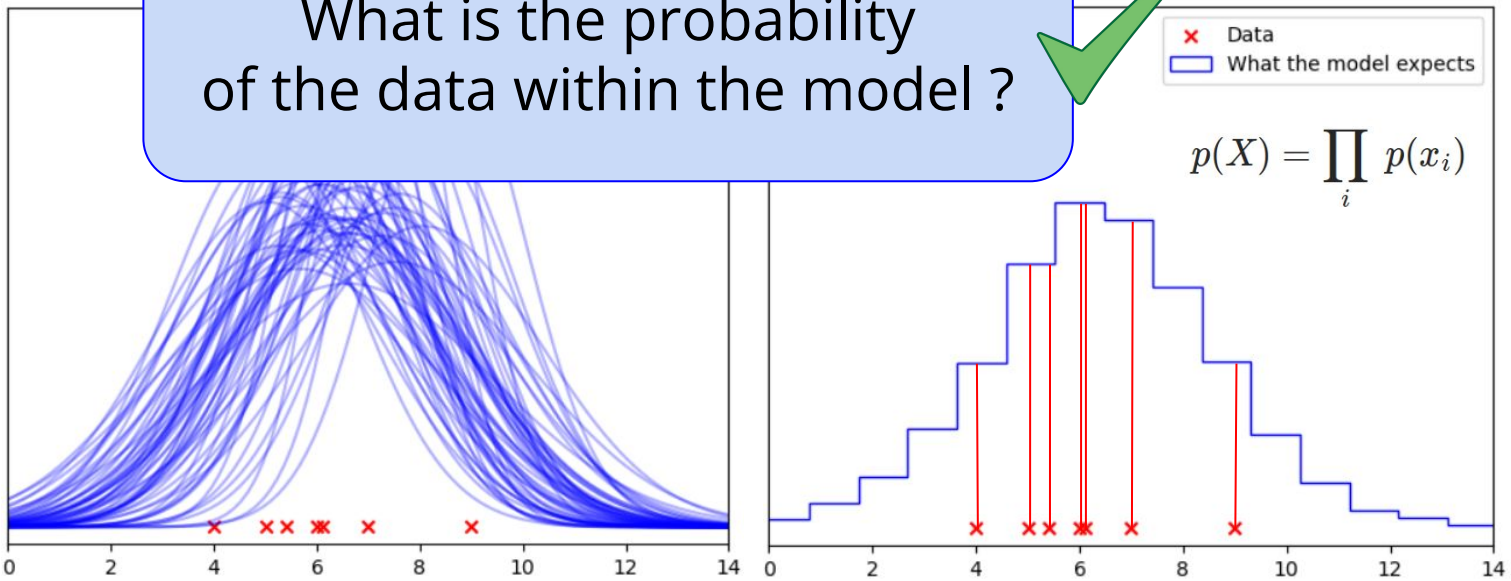




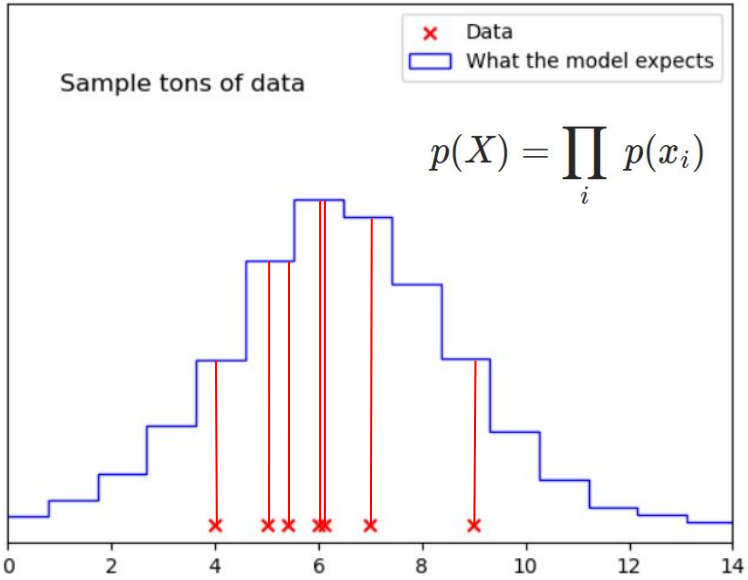
What is the probability of the data ?

- Data is usually multidimensional
- After inference we can easily sample tons of synthetic data!

What is the probability
of the data within the model ?

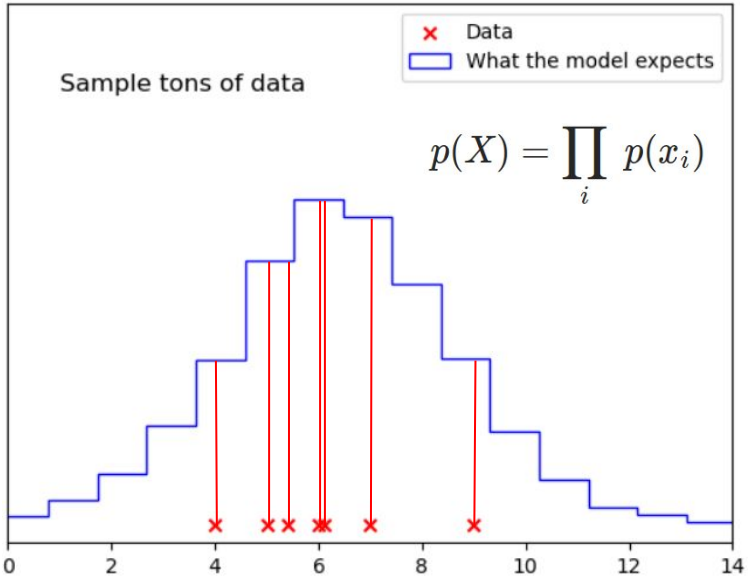


What is good and what is bad ?



p = 10^{-6} ... and now?

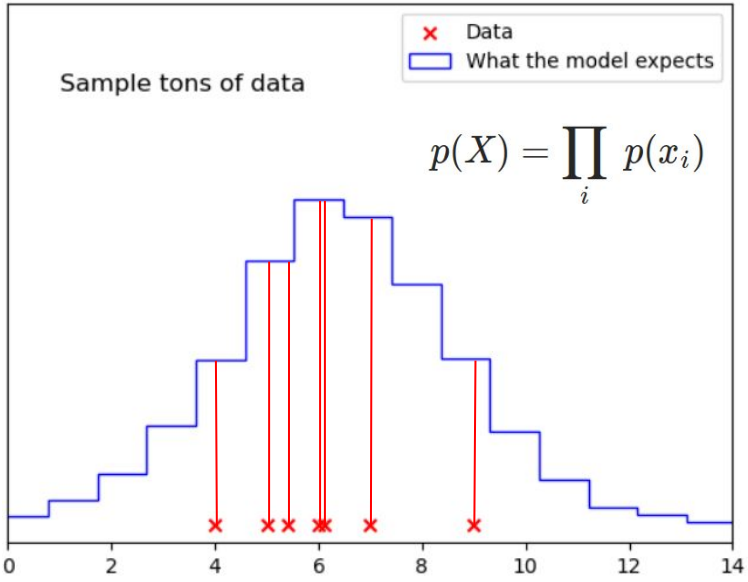
What is good and what is bad ?



p = 10^{-6} ... and now?

- What does it mean ?
- What do we compare it to ?

What is good and what is bad ?

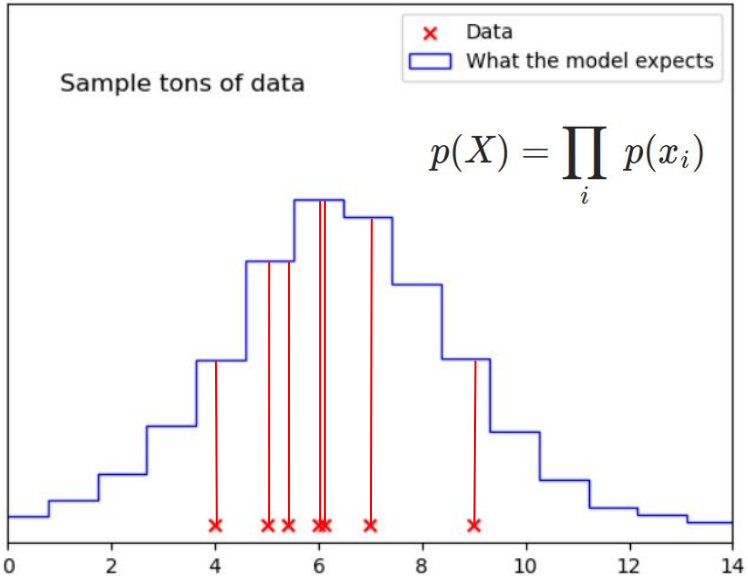


p = 10^{-6} ... and now?

- What does it mean ?
- What do we compare it to ?

- Generate replicas of data X^{Rep}

What is good and what is bad ?

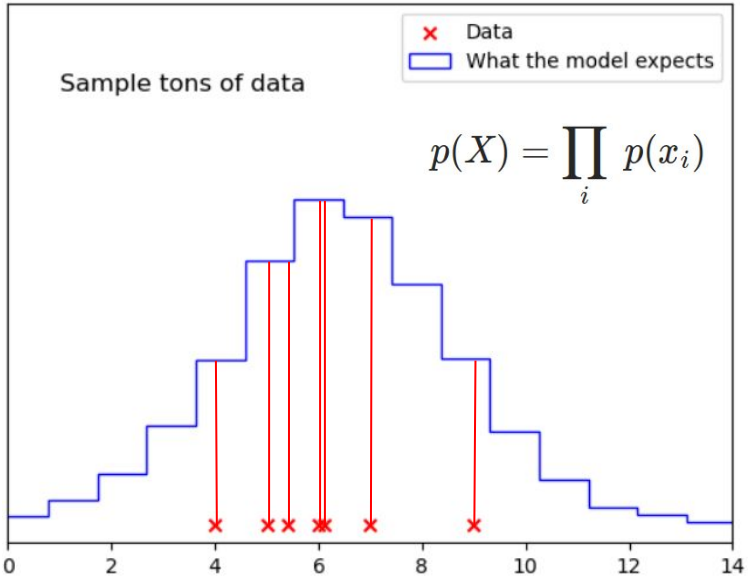


p = 10^{-6} ... and now?

- What does it mean ?
- What do we compare it to ?

- Generate replicas of data X^{Rep}
- Compute their probability

What is good and what is bad ?



p = 10^{-6} ... and now?

- What does it mean ?
- What do we compare it to ?

- Generate replicas of data X^{Rep}
- Compute their probability
- Compute

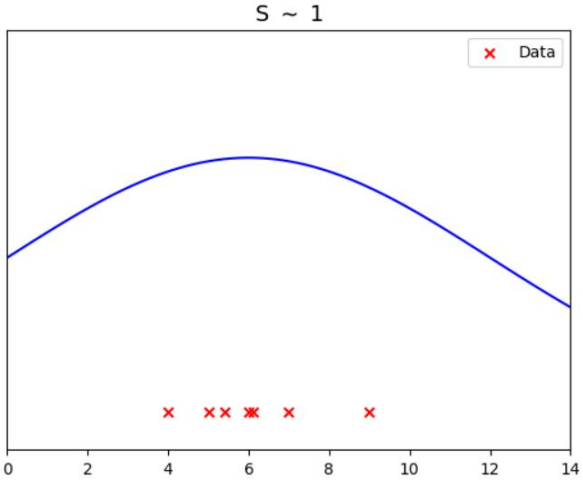
$$p\left(p(X^{rep}) < p(X) \right)$$

What is good and what is bad ?

$$\text{Score} = p\left(p(X^{rep}) < p(X) \right)$$

What is good and what is bad ?

$$\text{Score} = p\left(p(X^{rep}) < p(X) \right)$$

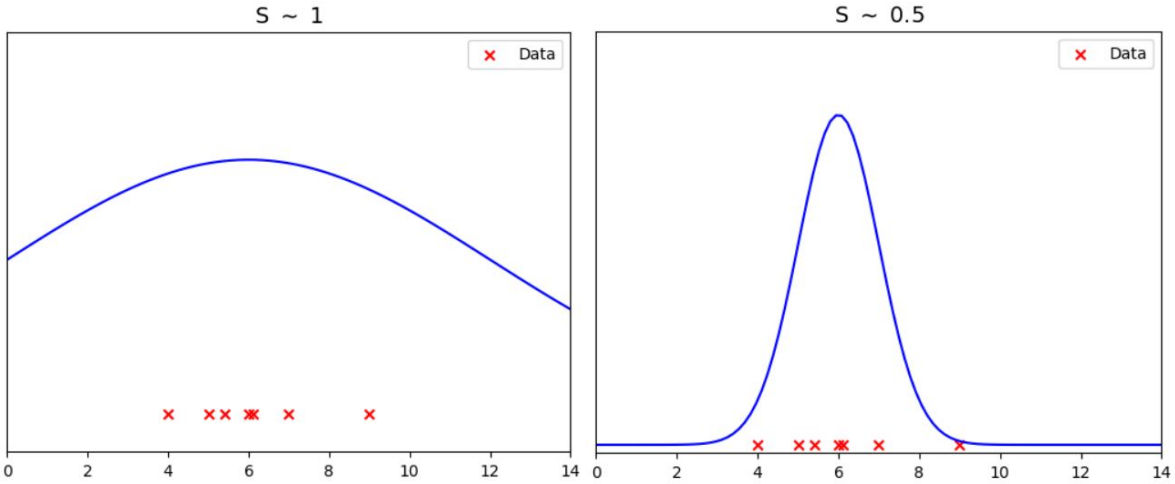


- S ~ 1 : bad model

What is good and what is bad ?

$$\text{Score} = p\left(p(X^{rep}) < p(X) \right)$$

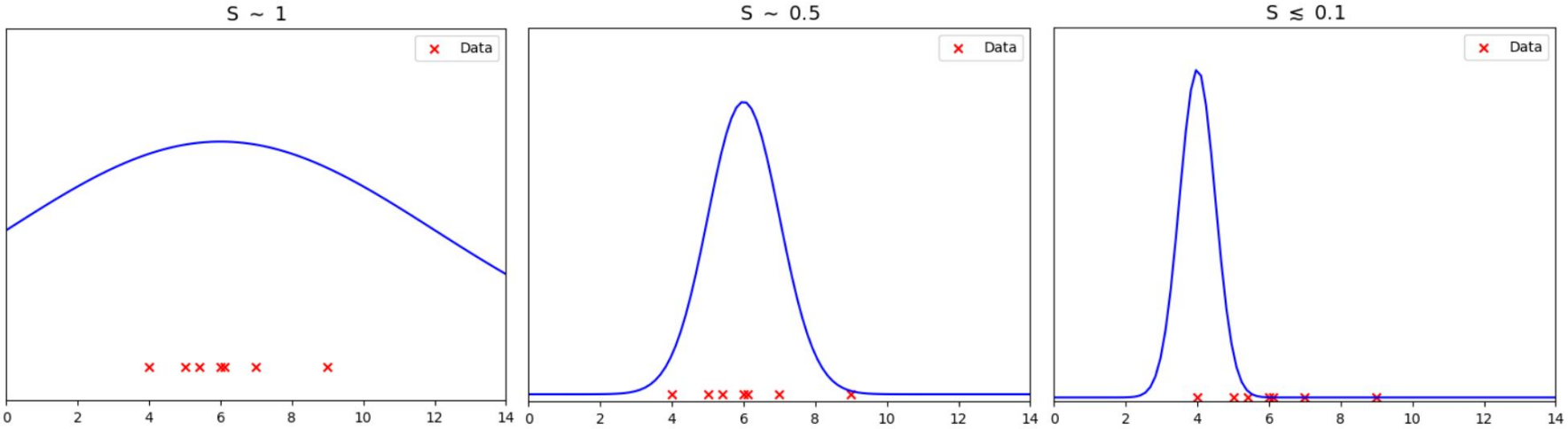
- S ~ 1 : bad model
- S ~ 0.5: good model



What is good and what is bad ?

$$\text{Score} = p\left(p(X^{rep}) < p(X) \right)$$

- S ~ 1 : bad model
- S ~ 0.5: good model
- S ≲ 0.1: bad model



What is good and what is bad ?

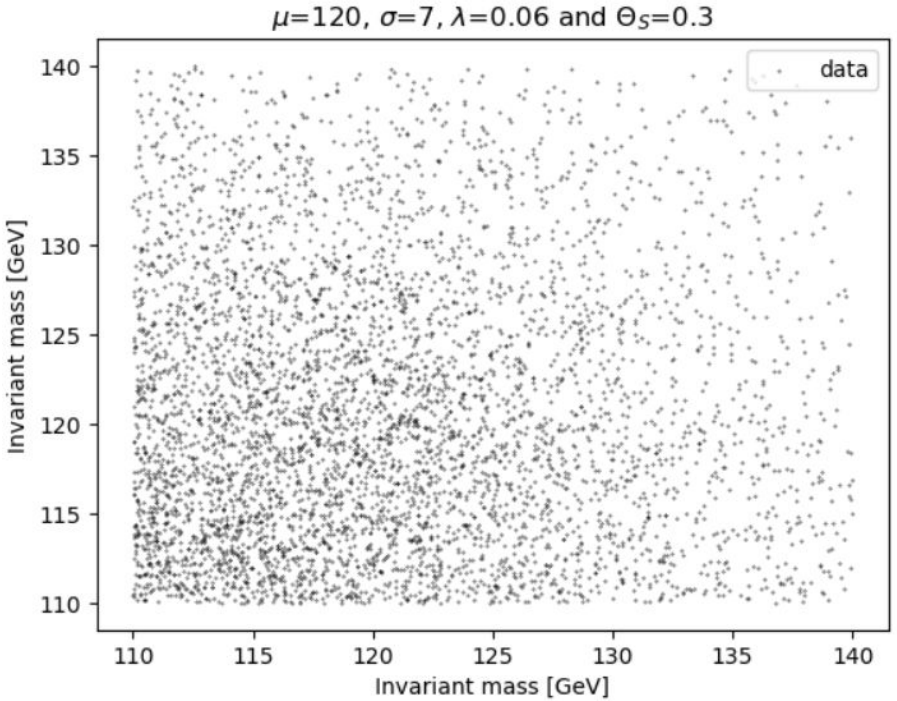
$$\text{Score} = p\left(p(X^{rep}) < p(X) \right)$$

Avoid overfit: given the data X, leave a held-out X_{held} for the posterior predictive check!

$$\text{Score} = p\left(p(X^{rep}_{held}) < p(X_{held}) \right)$$

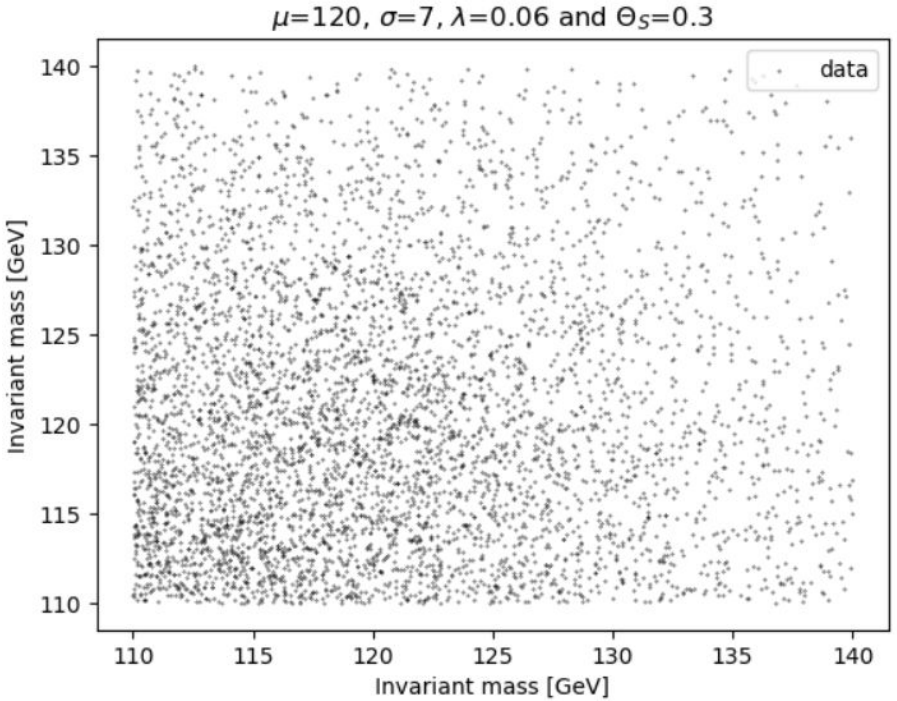
Realistic 2D example

Posterior Predictive Check: pp → bbγγ

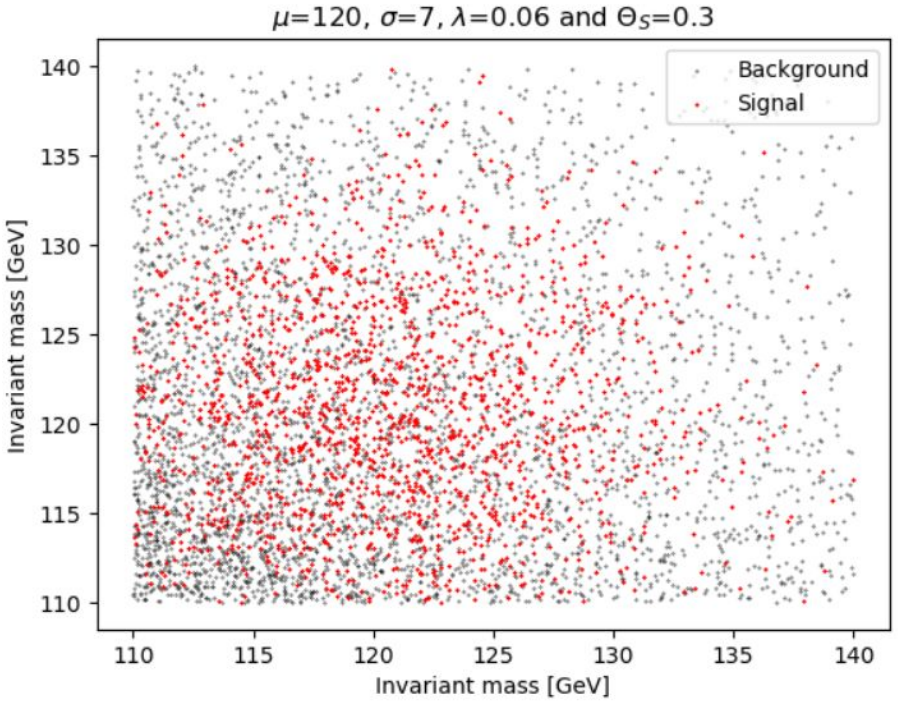


Data

Posterior Predictive Check: pp → bbγγ

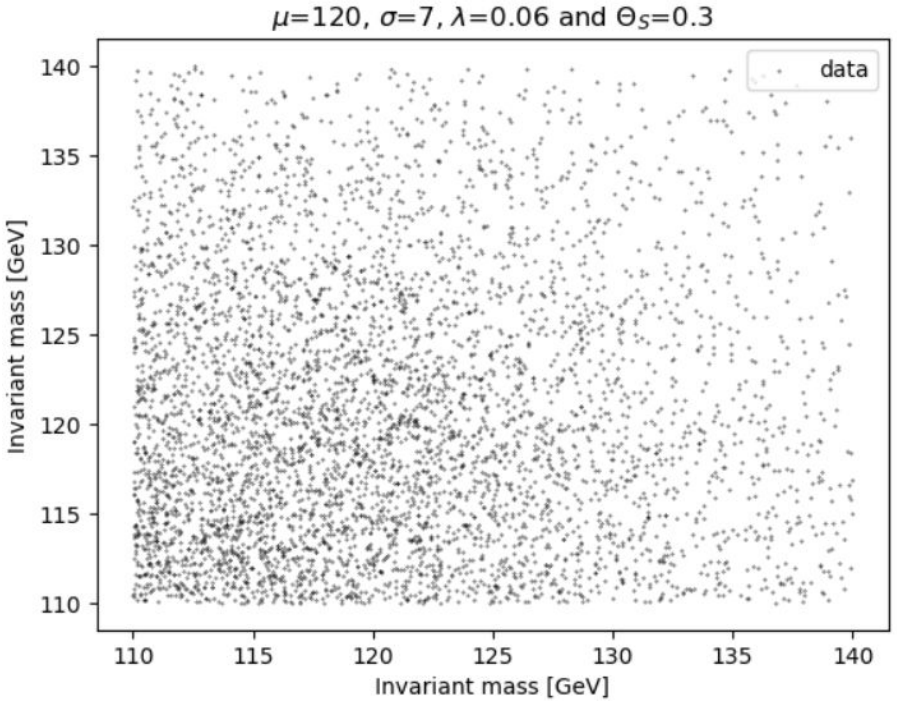


Data

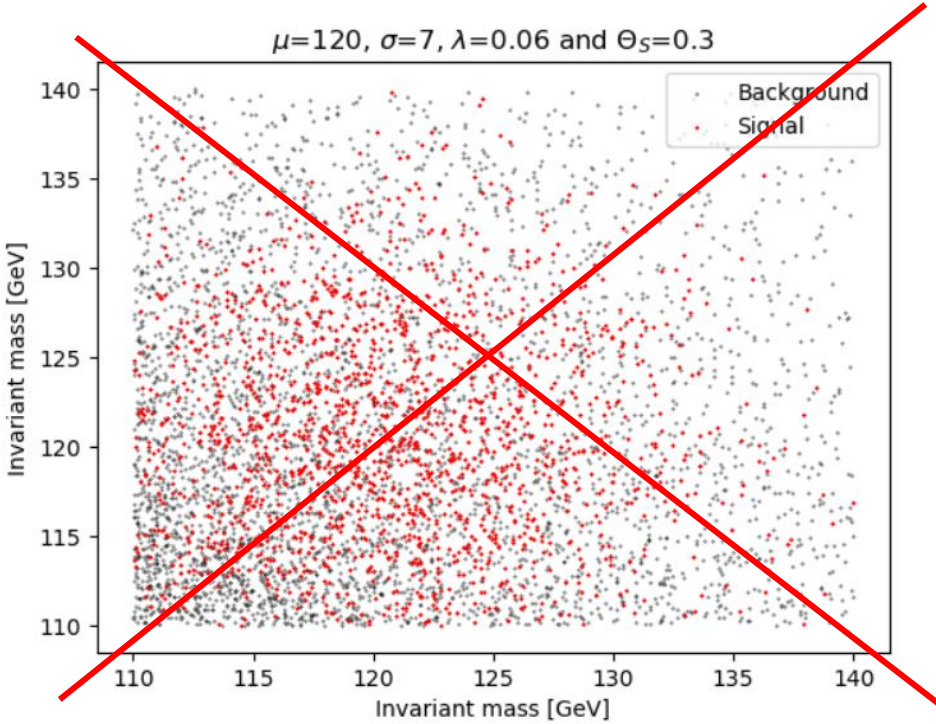


Labelled Data

Posterior Predictive Check: pp → bbγγ

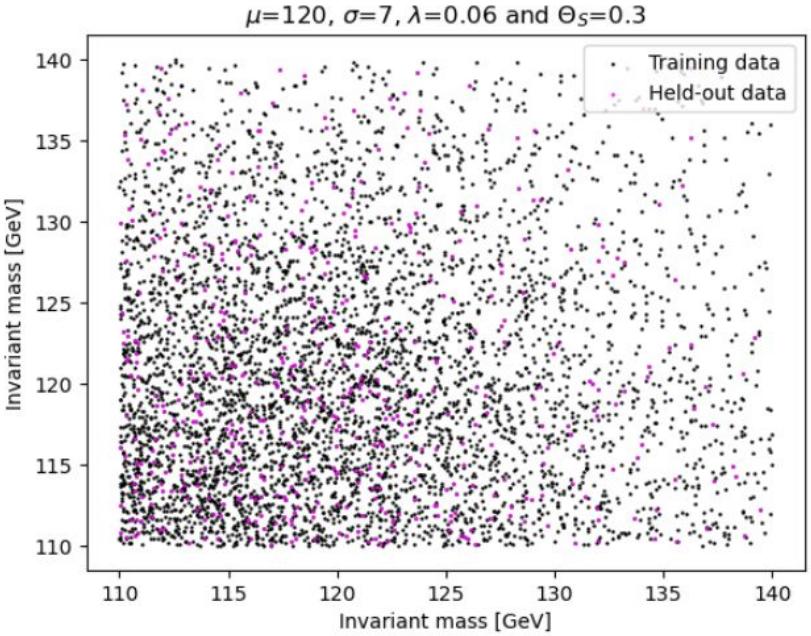


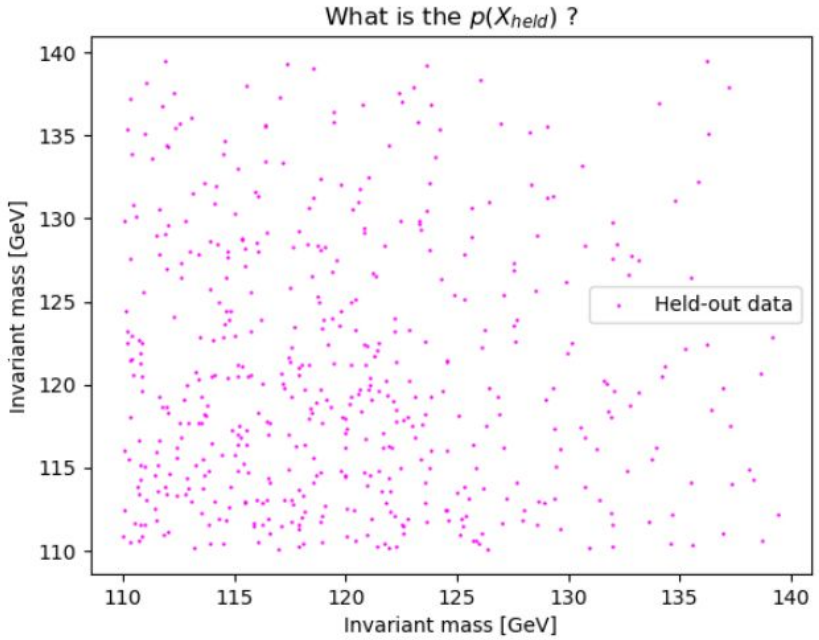
Data



Labelled Data

Posterior Predictive Check: pp → bbγγ

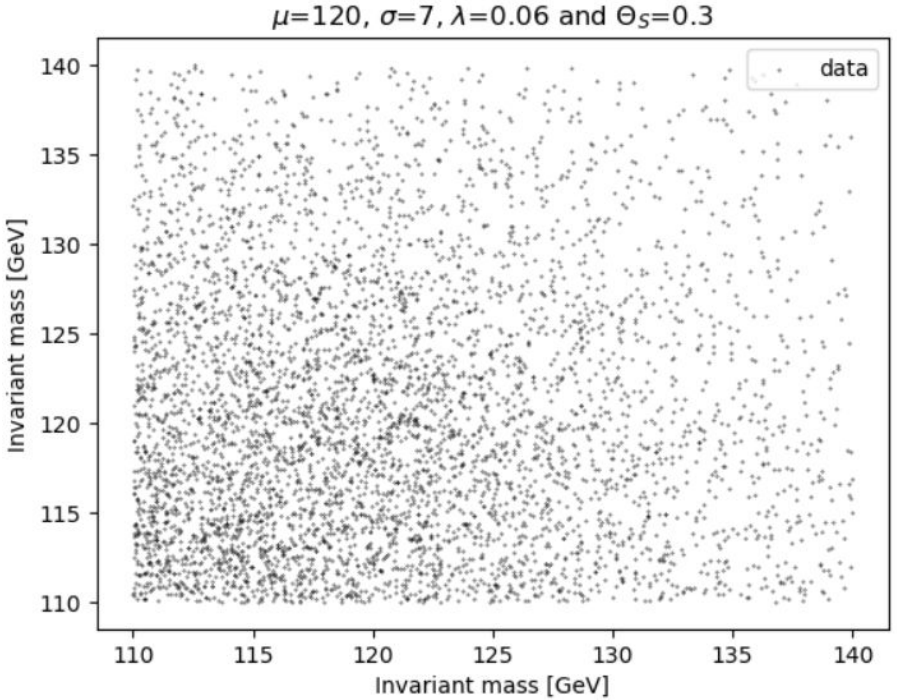




Data: 80% to infer, 20% held-out

Held-out data

Posterior Predictive Check: pp → bbγγ

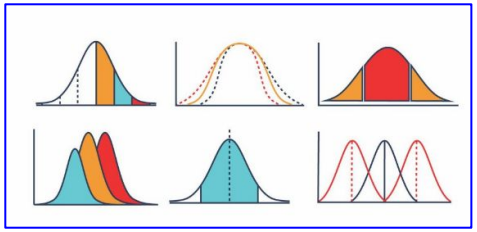


80% of data

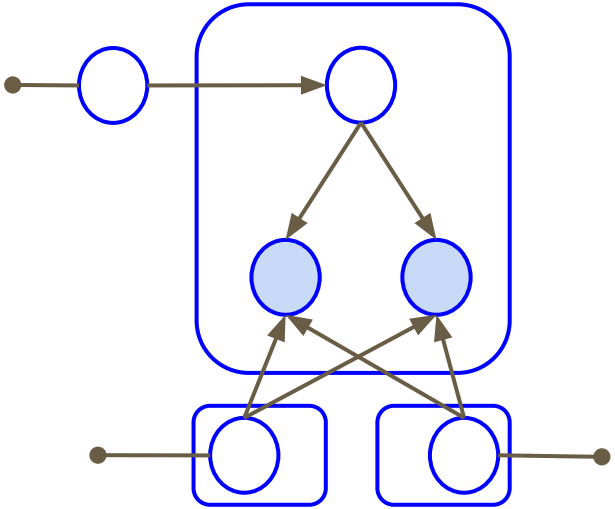
```
my_model = """
data {
 int <lower=0> N; // number of datapoints
 real L, U; // lower and upper limits of the observables (i.e. how to truncate the distributions)
 array[N,2] real<lower=L, upper=U> y;  // there are 2 observables per datapoint
 real mu0, sigma0, mu1, sigma1, mu2, sigma2, t1, t2;  // hyperparameters for the parameters priors
}

parameters {
  real<lower=0> mu;
  real<lower=0> sigma;
  real<lower=0> lambda0;
  simplex[2] theta;
}

model {
    vector[2] lp;
    mu ~ normal(mu0, sigma0);
    sigma ~ cauchy(mu1, sigma1);
    lambda0 ~ normal(mu2, sigma2);
    theta ~ dirichlet([t1,t2]);

   for (n in 1:N) {
    lp[1] =   exponential_lpdf( y[n,1] - L | lambda0)
            - exponential_lcdf( U - L | lambda0)
            + exponential_lpdf( y[n,2] - L| lambda0)
            - exponential_lcdf( U - L | lambda0);
     lp[2] =  normal_lpdf(y[n,1] | mu, sigma)
            - log_diff_exp( normal_lcdf( U | mu, sigma), normal_lcdf( L | mu, sigma) )
            + normal_lpdf(y[n,2] | mu, sigma)
            - log_diff_exp( normal_lcdf( U | mu, sigma), normal_lcdf( L | mu, sigma) );

     // Pay attention here: theta[1] is associated to background, and theta[2] to signal, since we don't have theta[0]
     target += log_mix(theta, lp);
     };
}
"""
```

We infer with our model

Posterior Predictive Check: pp → bbγγ

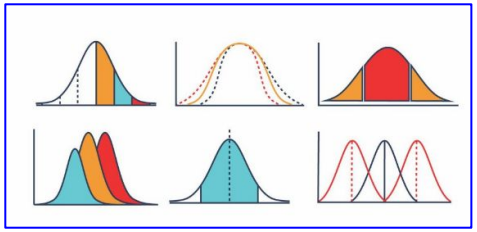


X

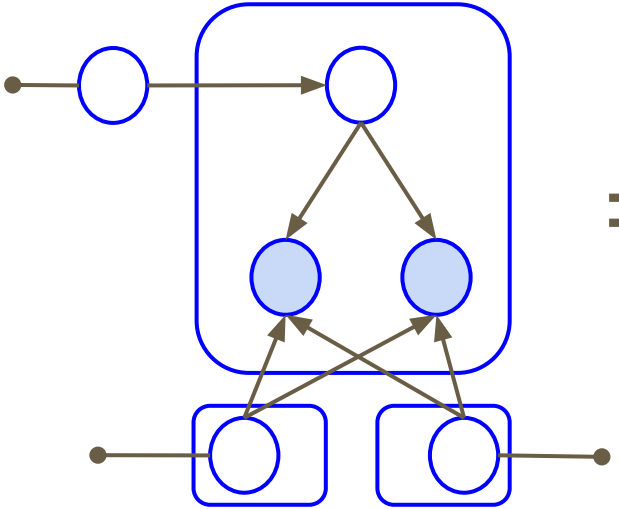


Posterior x model

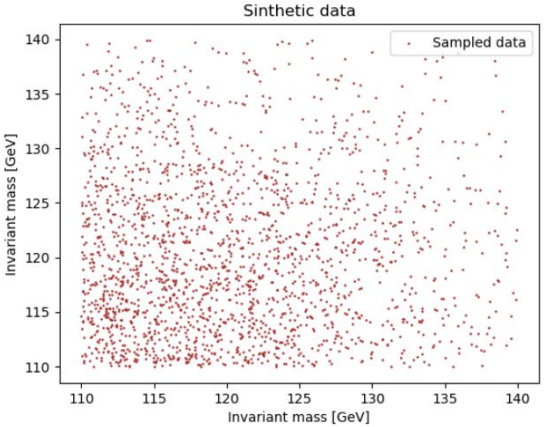
Posterior Predictive Check: pp → bbγγ



X

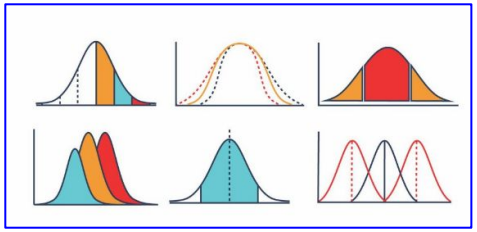


=

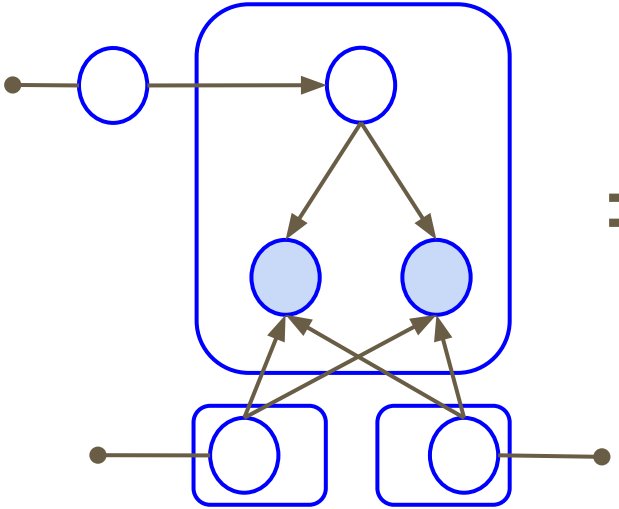


Posterior x model = sinthetic data !

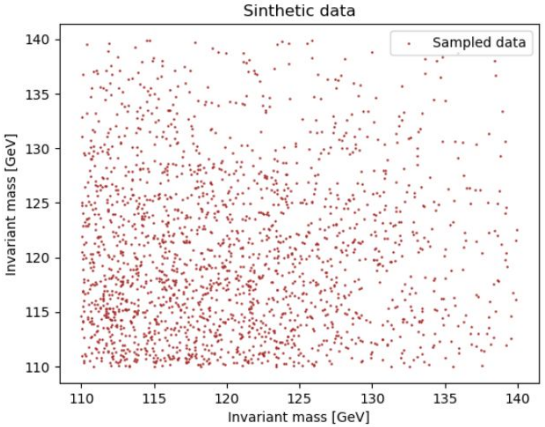
Posterior Predictive Check: pp → bbγγ



X

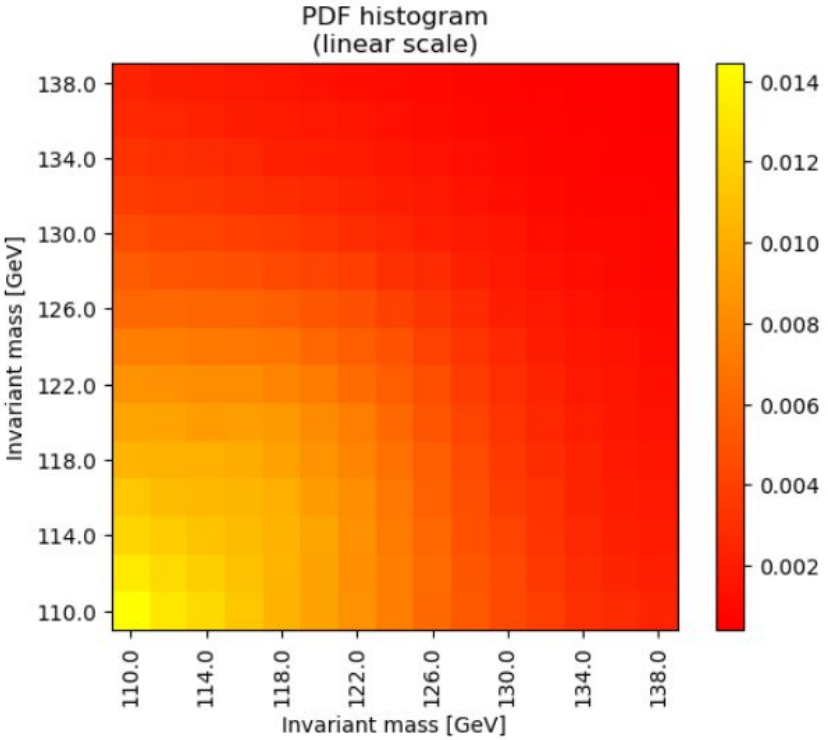


=



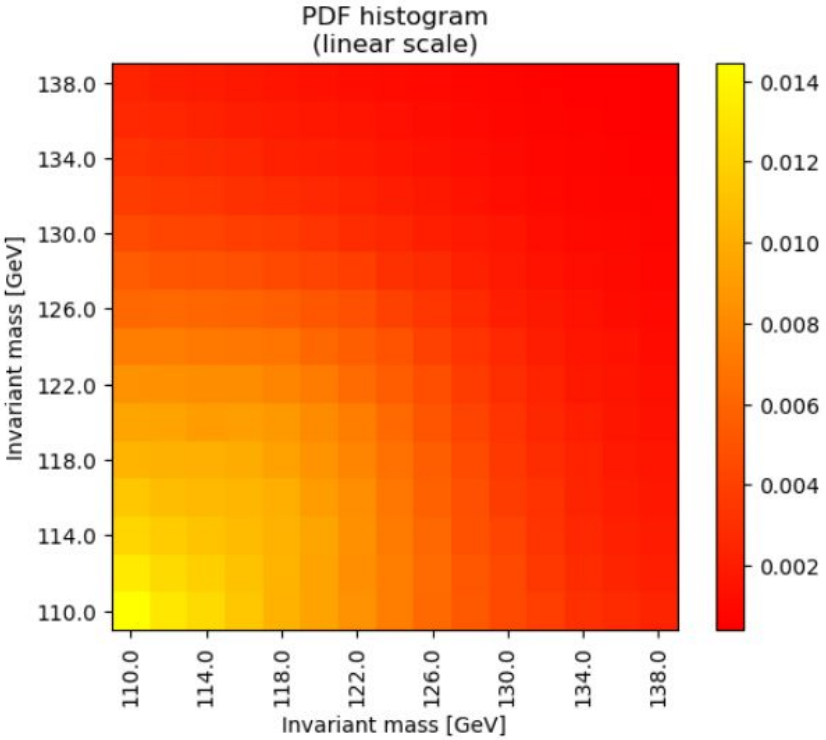
Posterior x model = Millions of sinthetic data !

PDF of your model given the data !



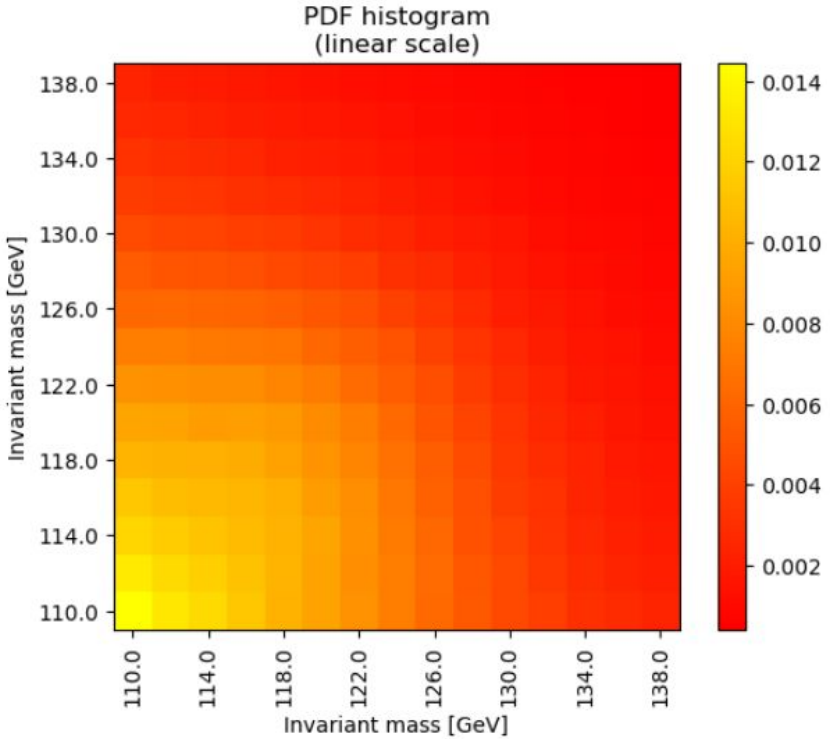
$$p(x_n|X)$$

PDF of your model given the data !



$$p(x_n|X) = \int p(x_n|z_i)\, p(z_i|X)\, dz_i$$

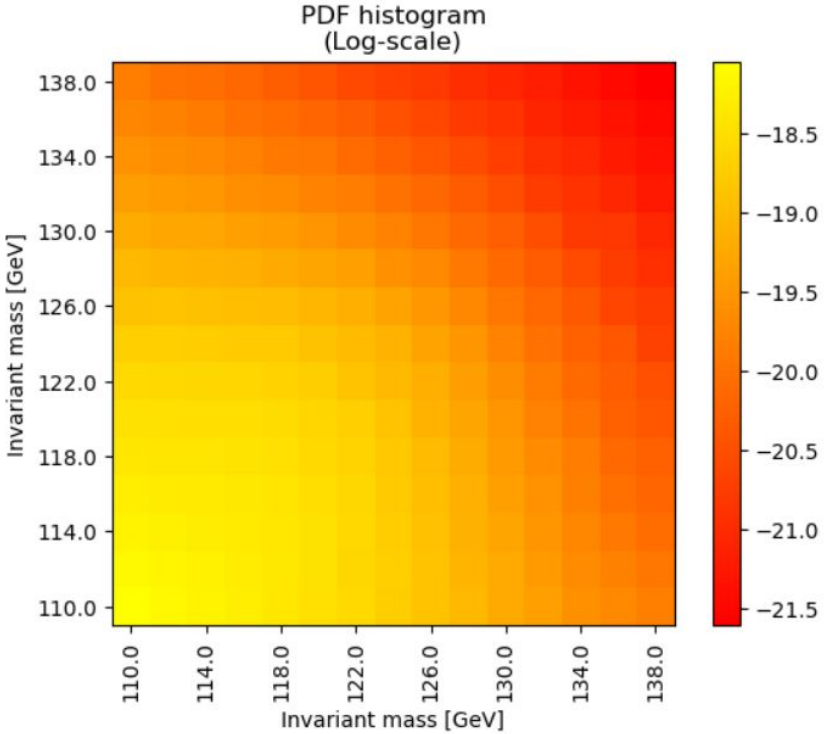
PDF of your model given the data !



$$p(x_n|X) = \int p(x_n|z_i)\, p(z_i|X)\, dz_i$$

We've never seen a label!

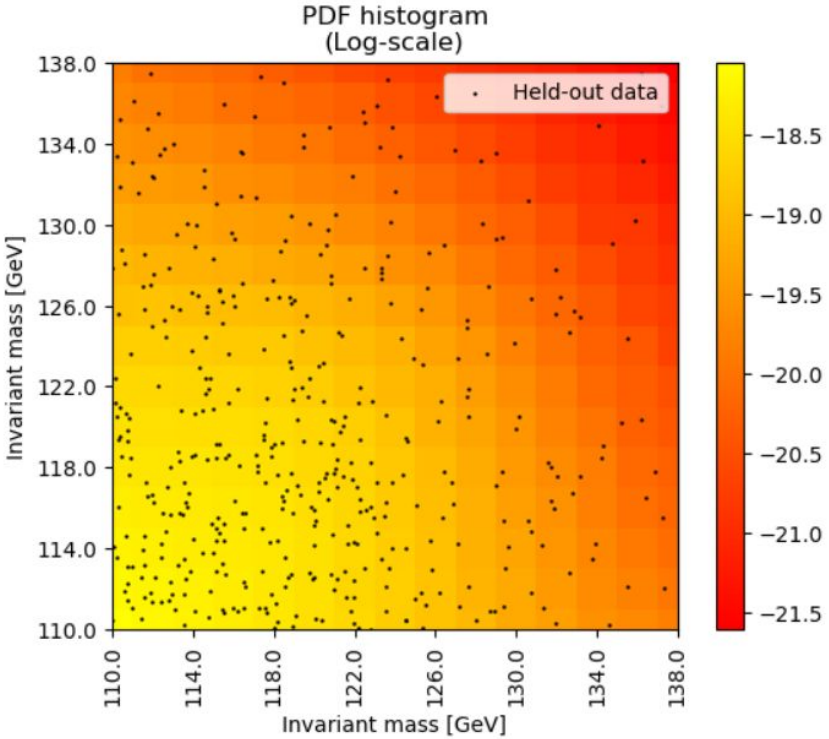
PDF of your model given the data !



$$p(x_n|X) = \int p(x_n|z_i)\, p(z_i|X)\, dz_i$$

Log-scale!

PDF of your model given the data !

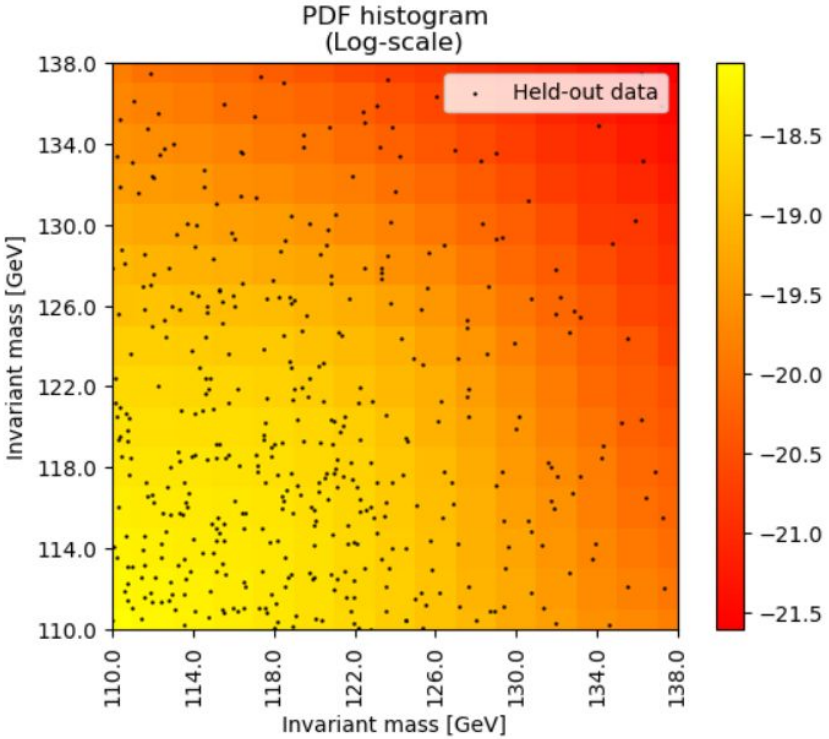


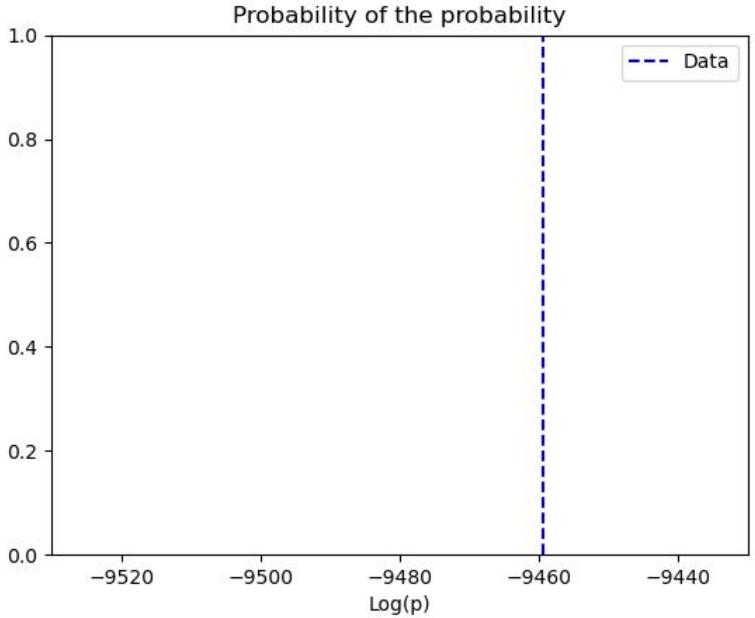
$$p(x_n|X) = \int p(x_n|z_i)\, p(z_i|X)\, dz_i$$

Log-scale!

Simply add the value of each bin for each held-out datapoint!

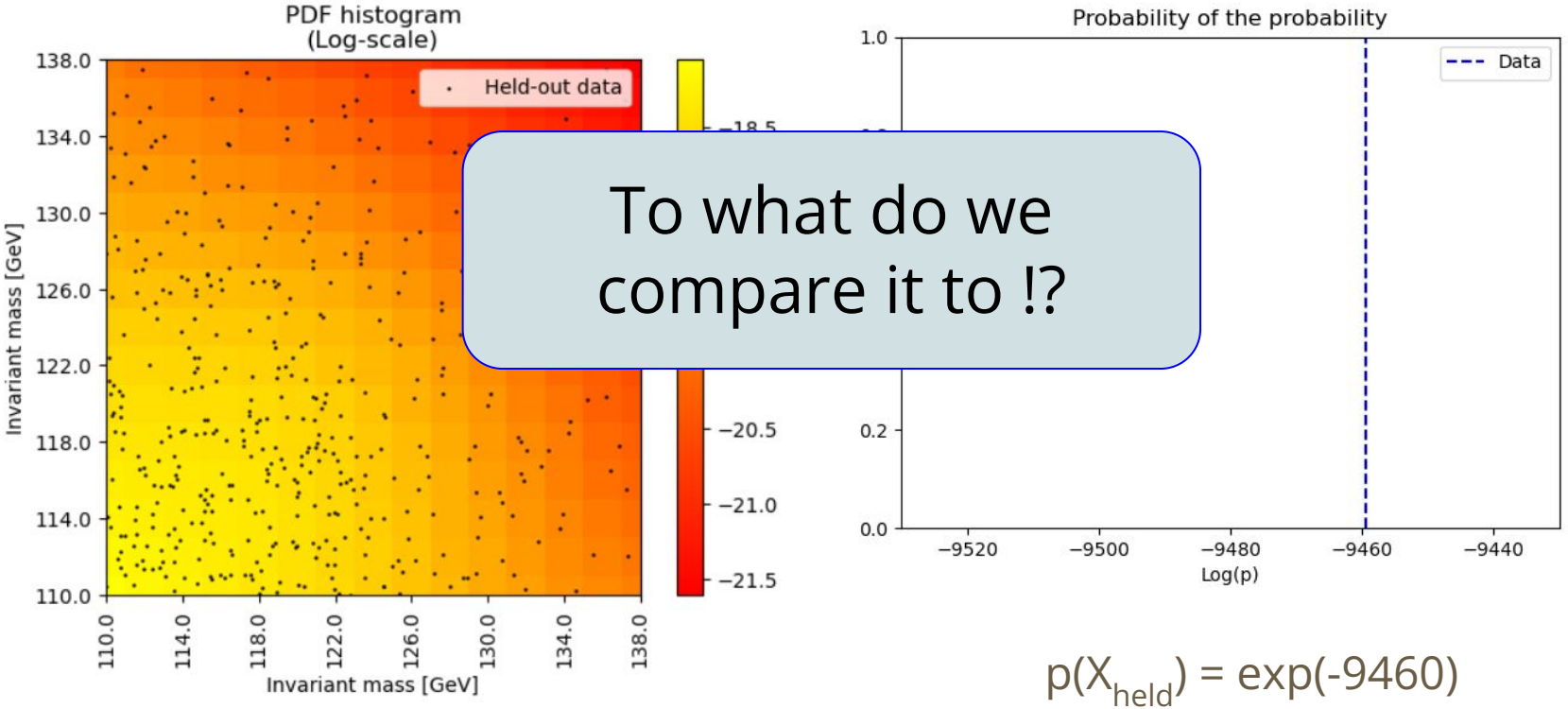
PDF of your model given the data !





$p(X_{held}) = \exp(-9460)$

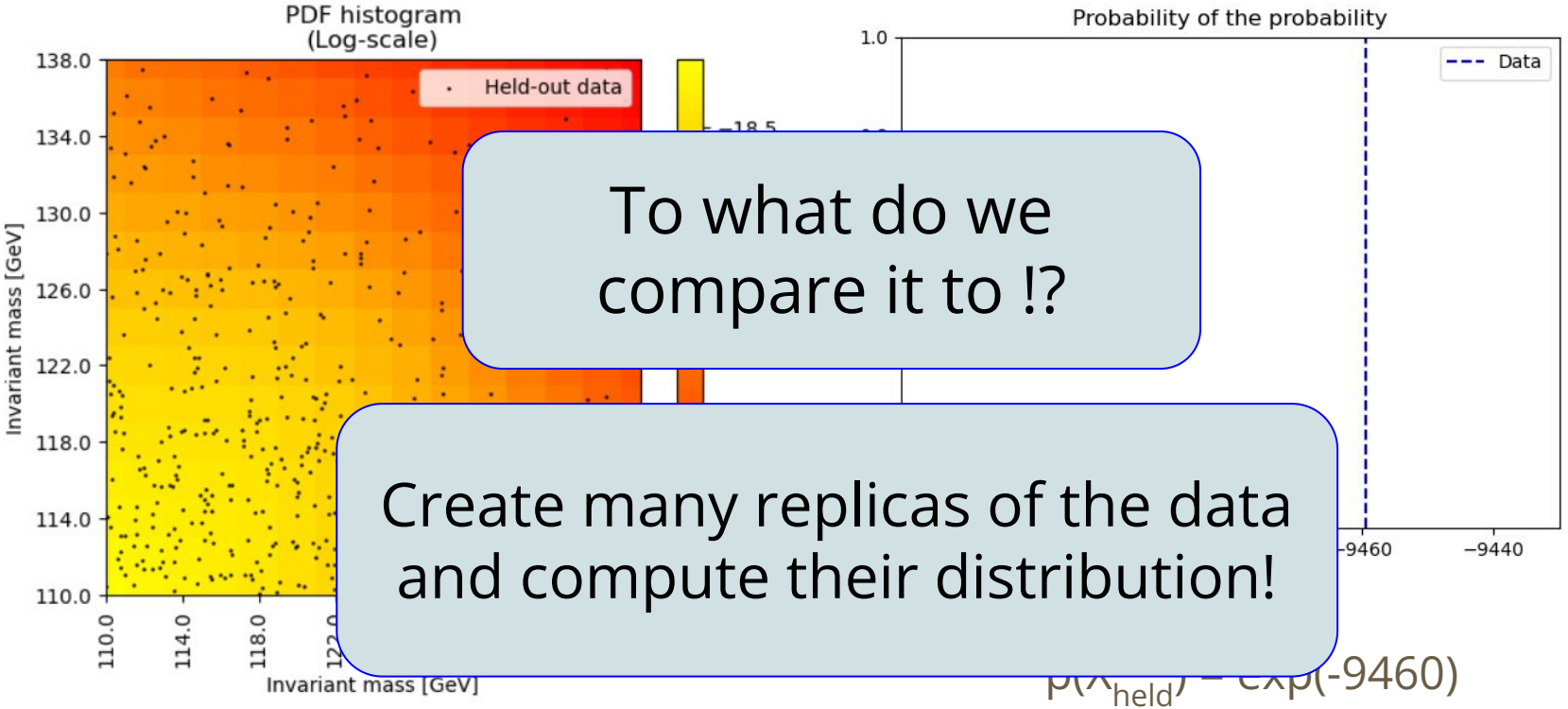
PDF of your model given the data !



To what do we compare it to !?

$p(X_{held}) = \exp(-9460)$

PDF of your model given the data !

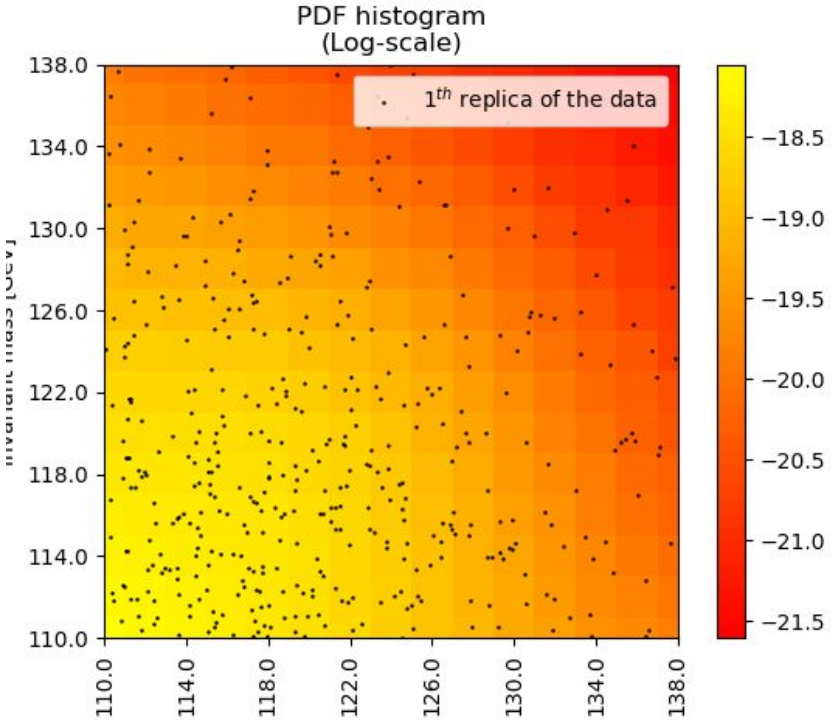


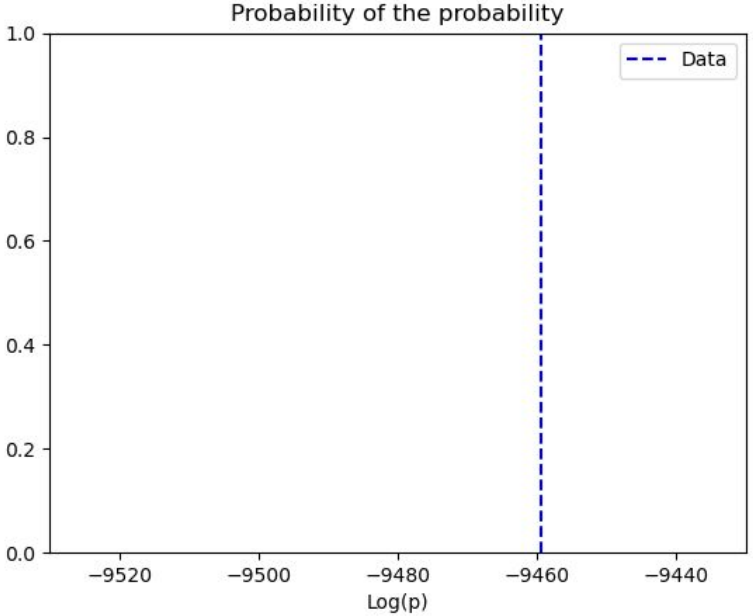
To what do we compare it to !?

Create many replicas of the data and compute their distribution!

$p(x_{held}) = \exp(-9460)$

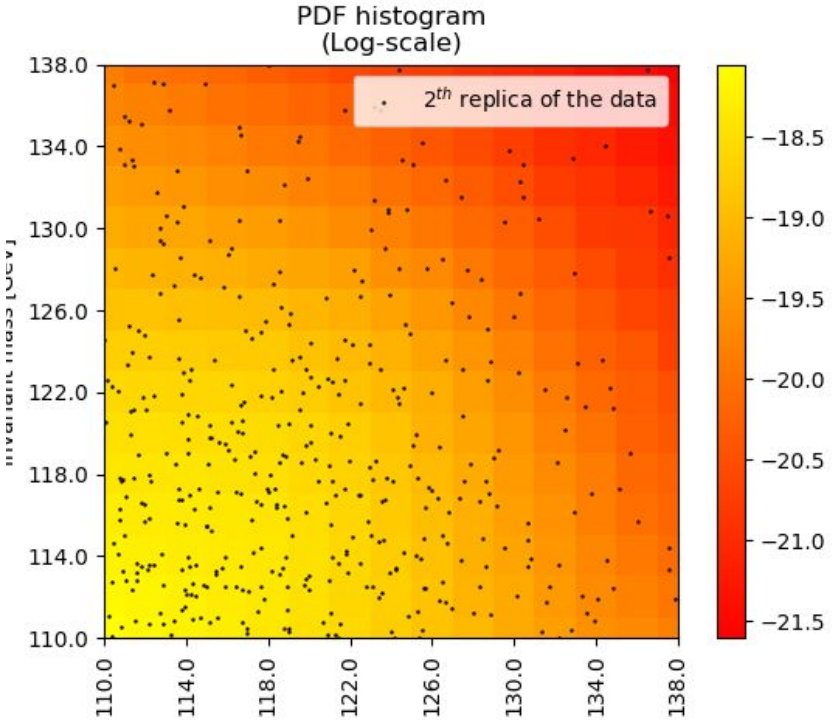
PDF of your model given the data !

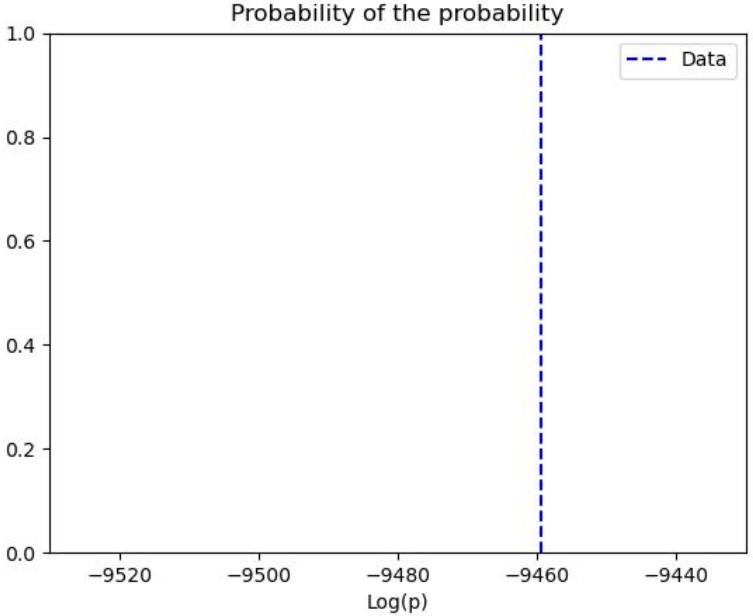




$p(X_{held}) = \exp(-9460)$

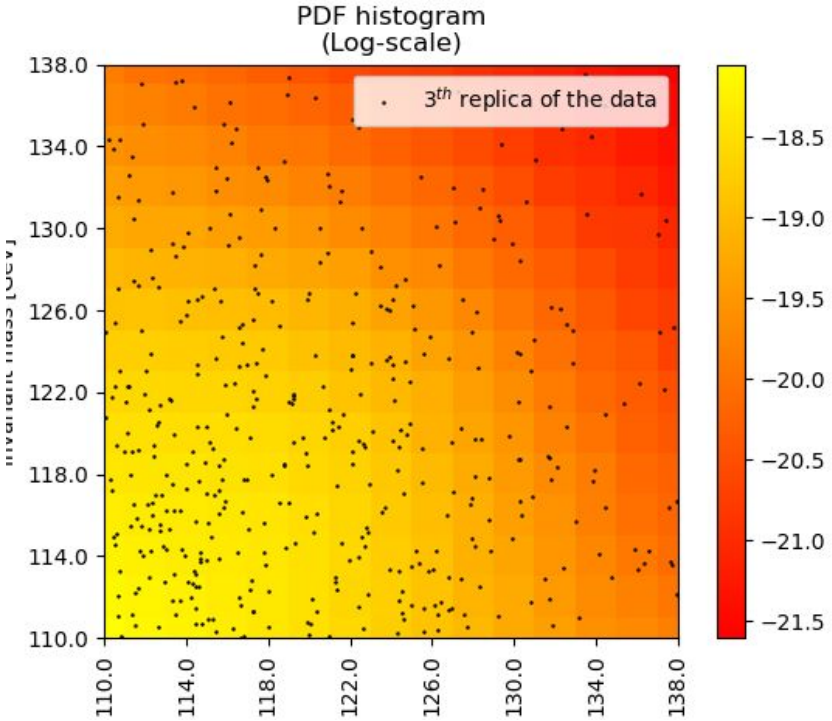
PDF of your model given the data !

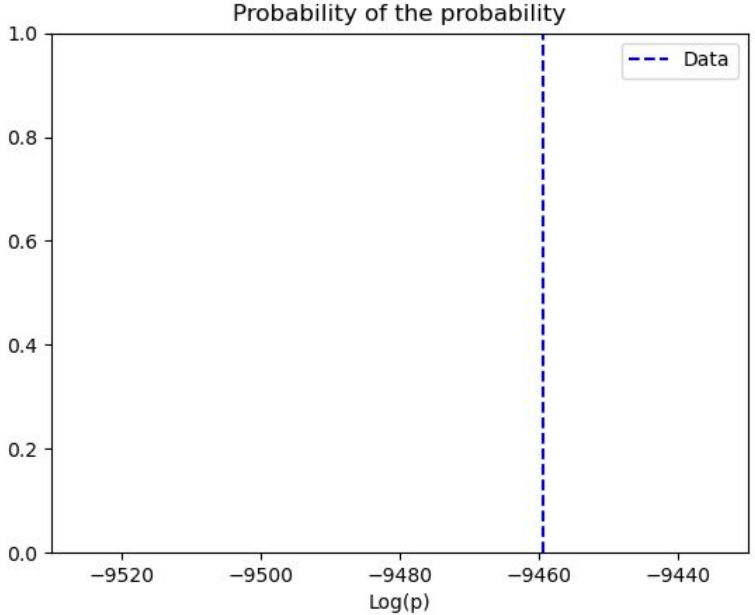




$p(X_{held}) = \exp(-9460)$

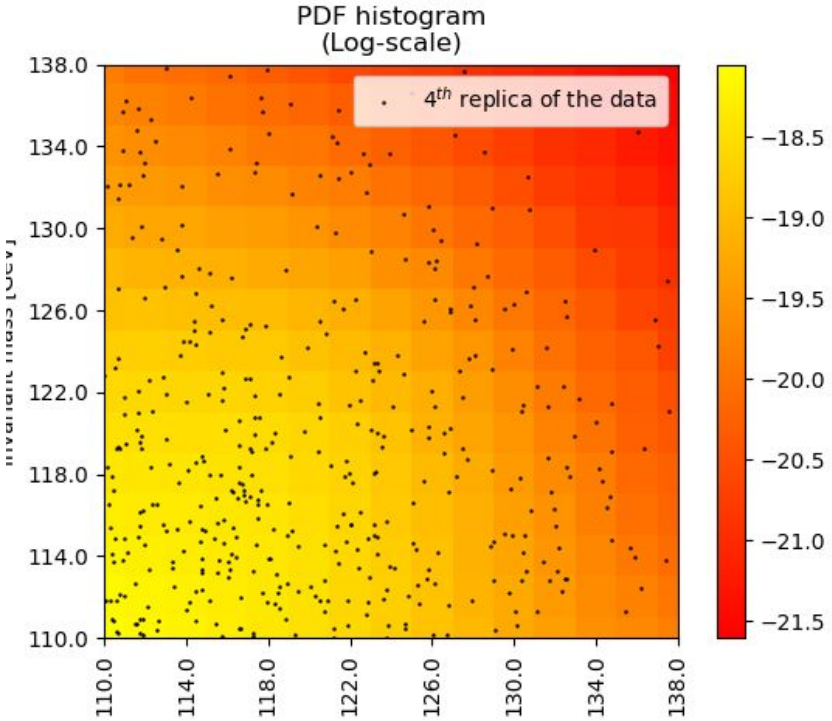
PDF of your model given the data !

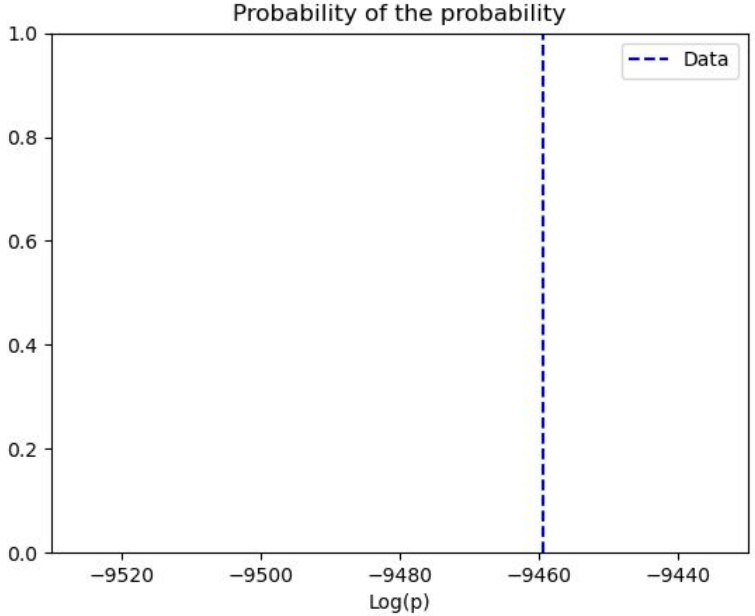




$p(X_{held}) = \exp(-9460)$

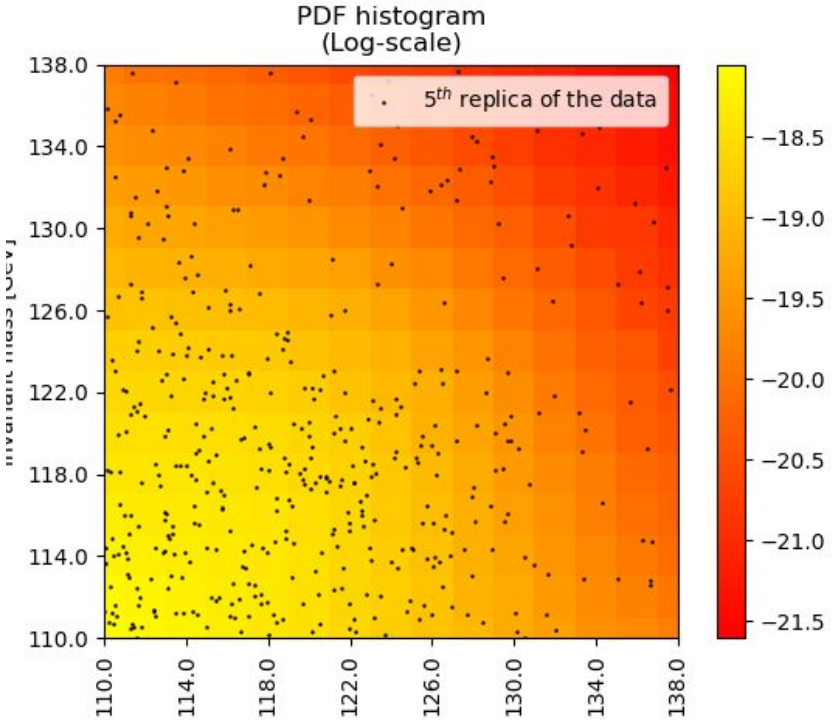
PDF of your model given the data !

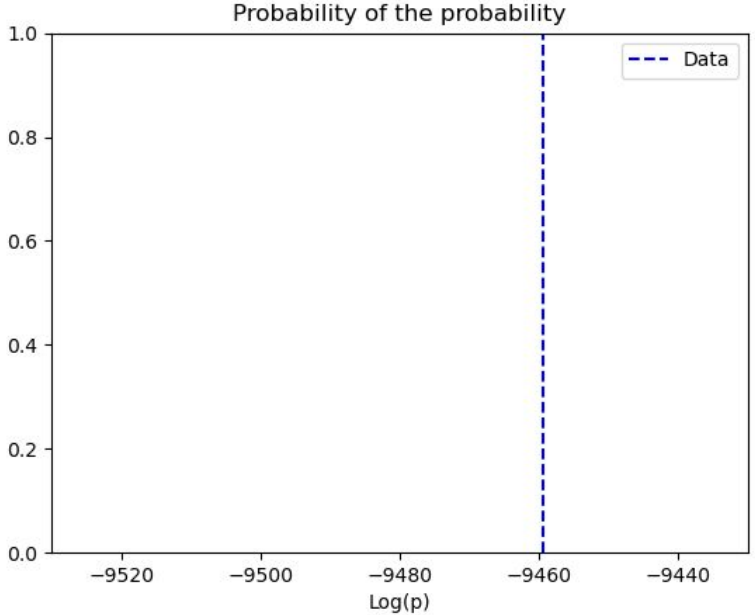




$p(X_{held}) = \exp(-9460)$

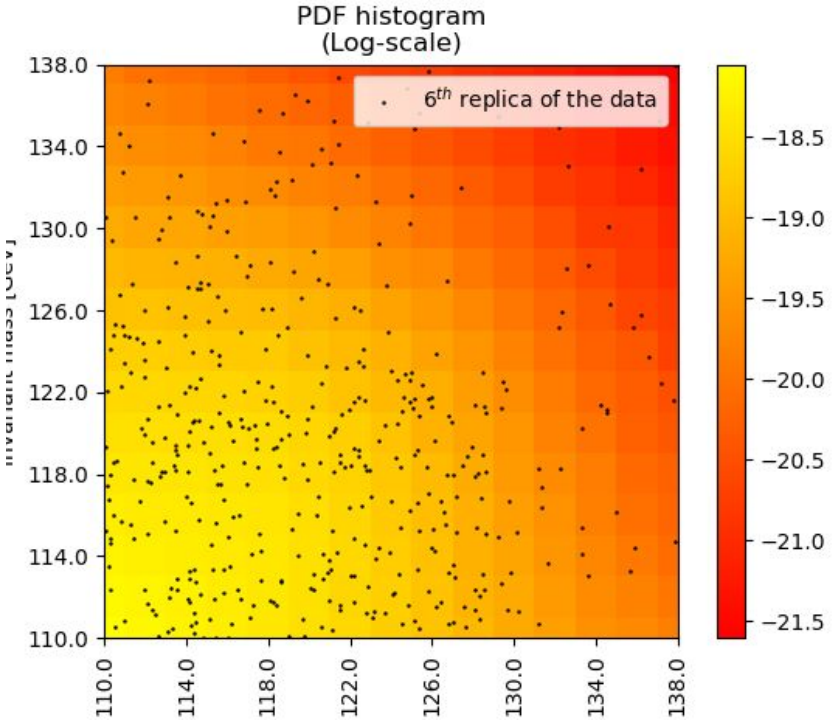
PDF of your model given the data !

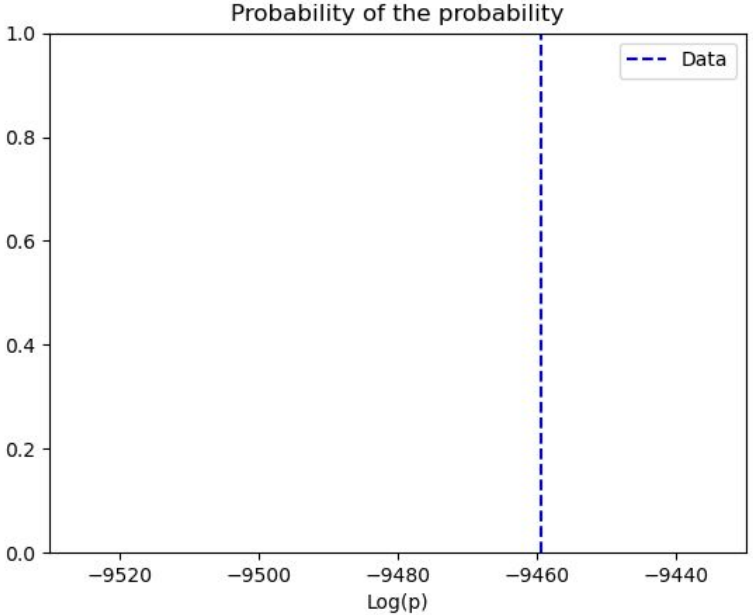




$p(X_{held}) = \exp(-9460)$

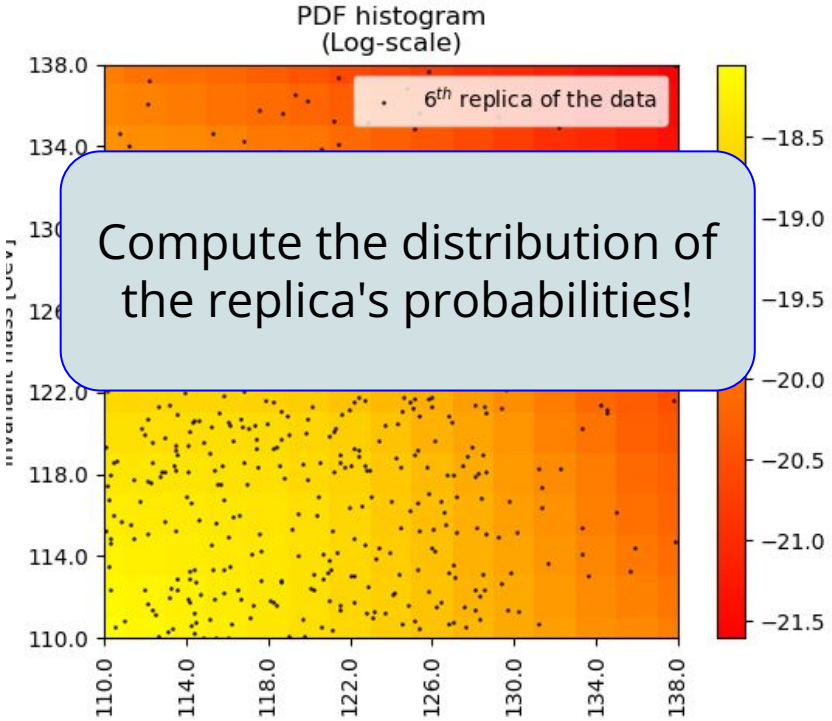
PDF of your model given the data !



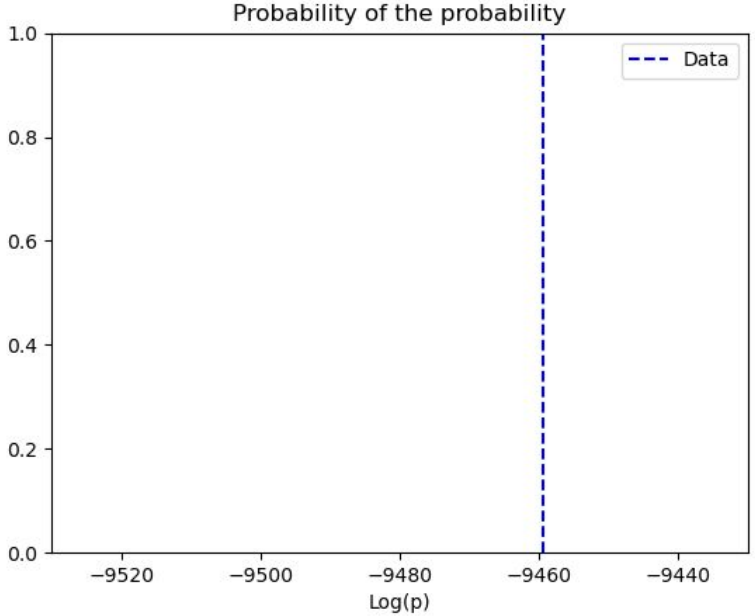


$p(X_{held}) = \exp(-9460)$

PDF of your model given the data !

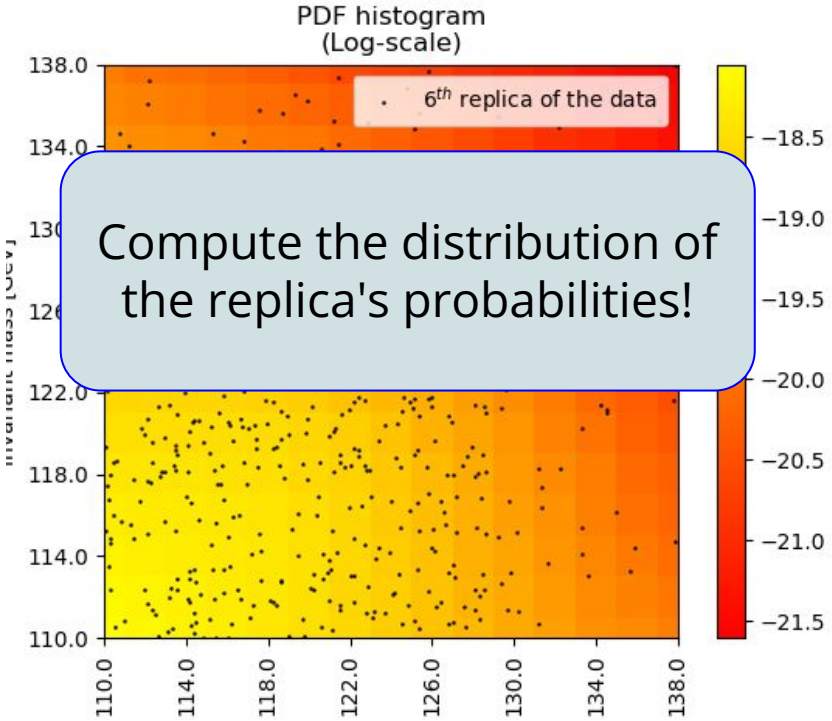


Compute the distribution of the replica's probabilities!

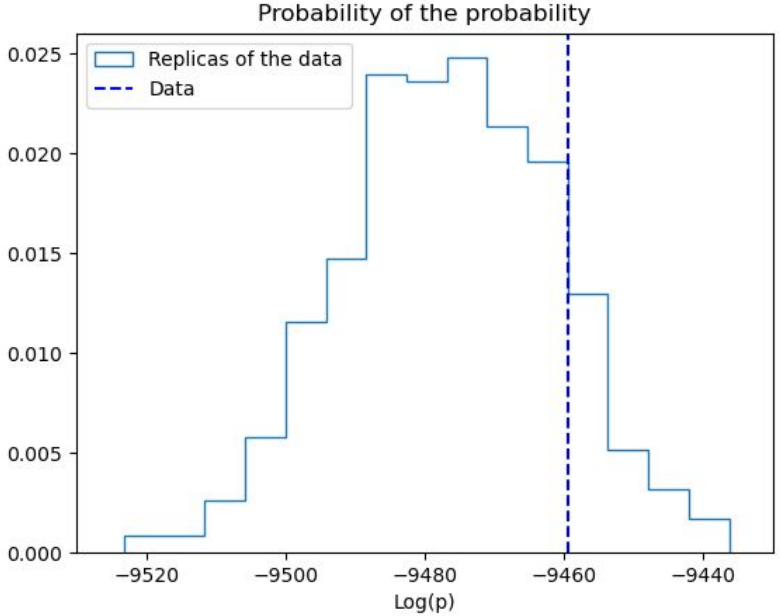


$p(X_{held}) = \exp(-9460)$

PDF of your model given the data !

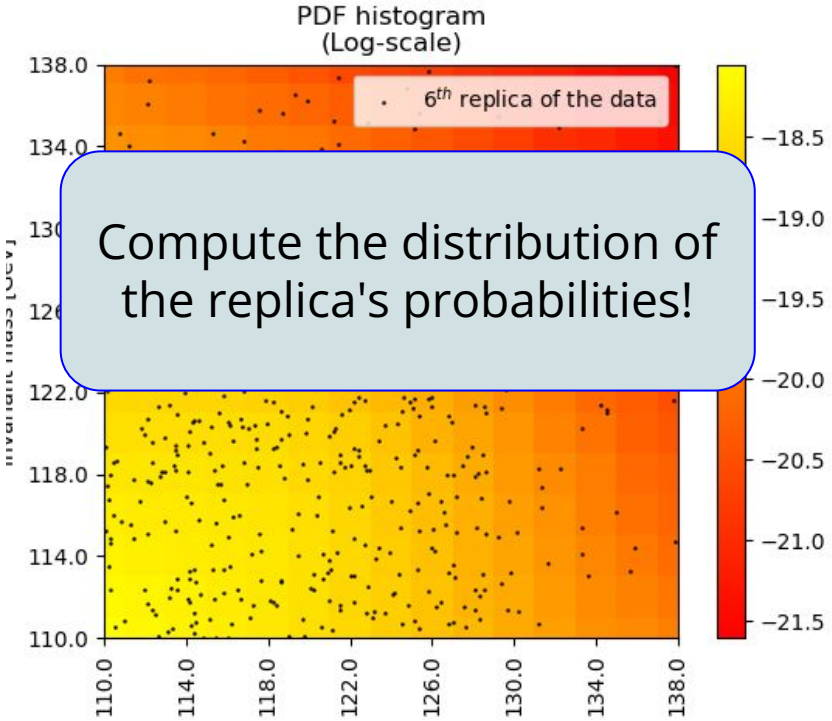


Compute the distribution of the replica's probabilities!

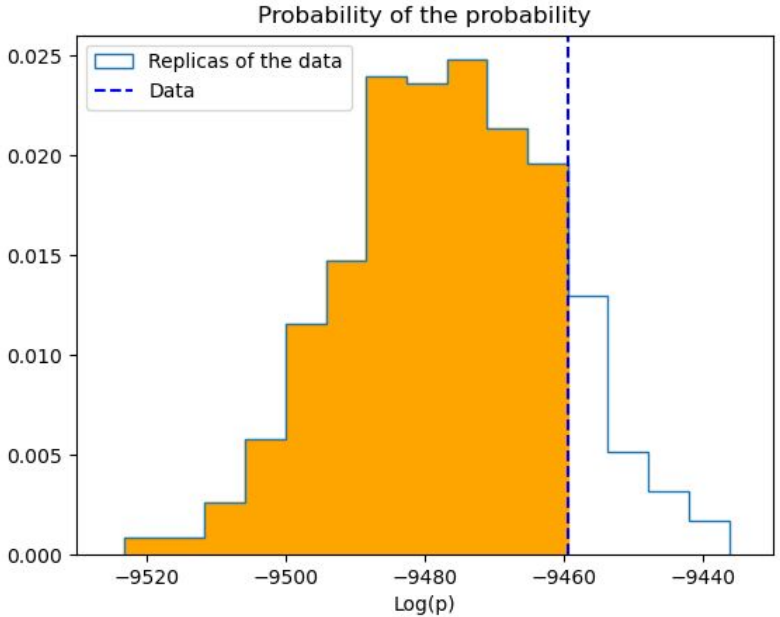


$p(X_{held}) = \exp(-9460)$

PDF of your model given the data !



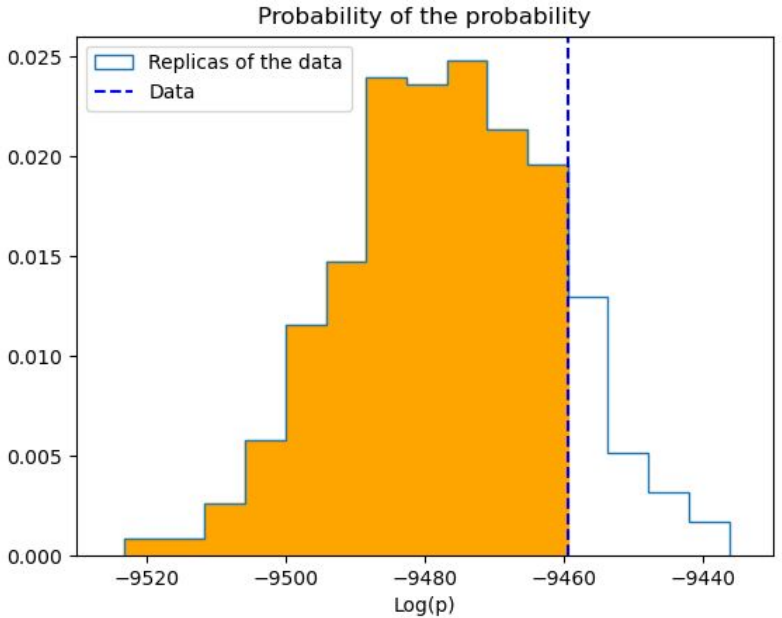
Compute the distribution of the replica's probabilities!



$$p\left(p(X_{held}^{rep}) < p(X_{held})\right) = 0.86$$

PDF of your model given the data !

Probability of X_{held} to have been sampled from the inferred model!



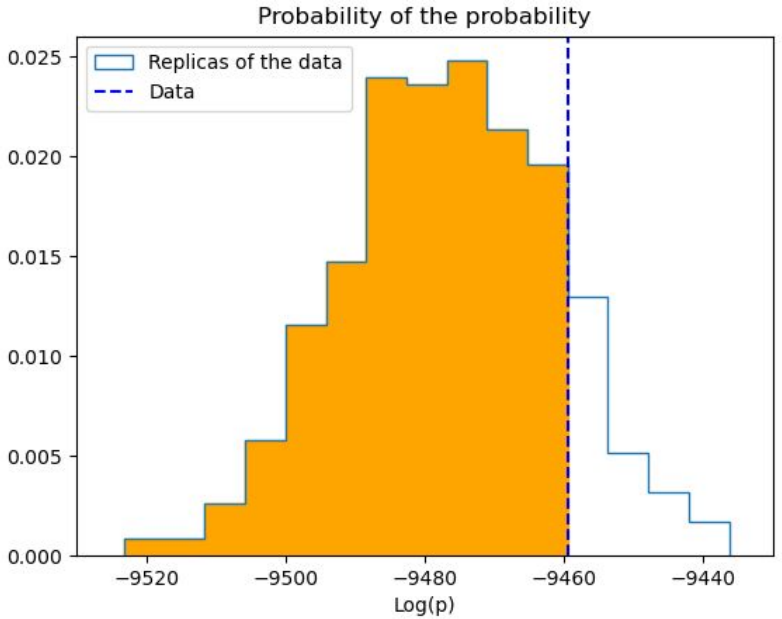
$$p\left(p(X_{held}^{rep}) < p(X_{held})\right) = 0.86$$

PDF of your model given the data !

Probability of X_{held} to have been sampled from the inferred model!

No general way:

Safeguard against gross misspecification



$$p\left(p(X_{held}^{rep}) < p(X_{held})\right) = 0.86$$

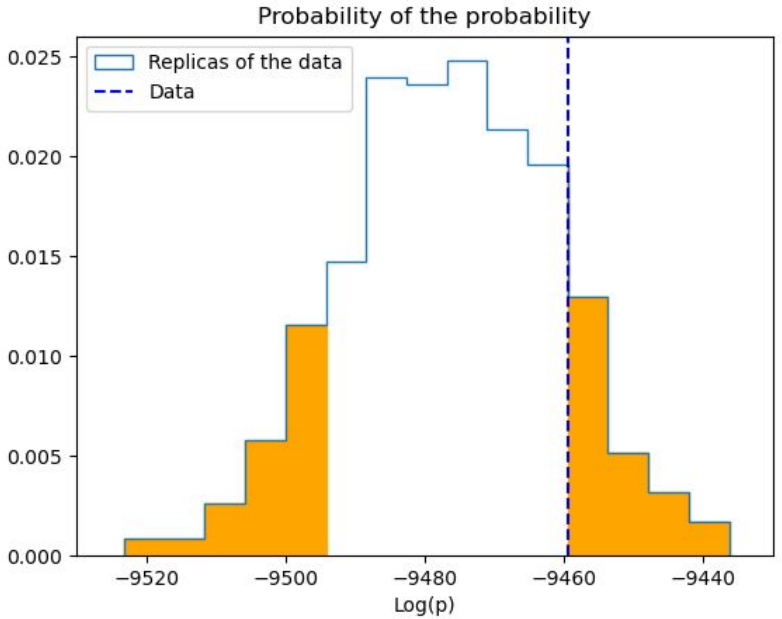
PDF of your model given the data !

Probability of X_{held} to have been sampled from the inferred model!

No general way:

Safeguard against gross misspecification

Another metric, all useful!



$$p\left(p^2(X_{held}^{rep}) < p^2(X_{held})\right) = 0.37$$

Final remarks @ Lecture 4

Assessment in Bayesian Machine Learning

Final remarks

- There are ways to check an inference process

Final remarks

- There are ways to check an inference process

  - Sampling diagnostics
  - Constructed data to find and understand problems
  - Posterior predictive checks

Final remarks

- There are ways to check an inference process
    - Sampling diagnostics
    - Constructed data to find and understand problems
    - Posterior predictive checks
- Bayesian Workflow: 2011.01808