Probabilistic & Approximate Computing

Sasa Misailovic

UIUC

Operations on Random Variables

$X \sim \mathcal{N}(0,1)$

$Y \sim \mathcal{N}(0,1)$

X and Y are i.i.d.

$Z = X + Y$ is also a Gaussian – $\mathcal{N}(0,2)$

Why?

Convolution!

$$\int_{-\infty}^{+\infty} f_X(x) \cdot f_Y(z - x) dx$$

But How About This?

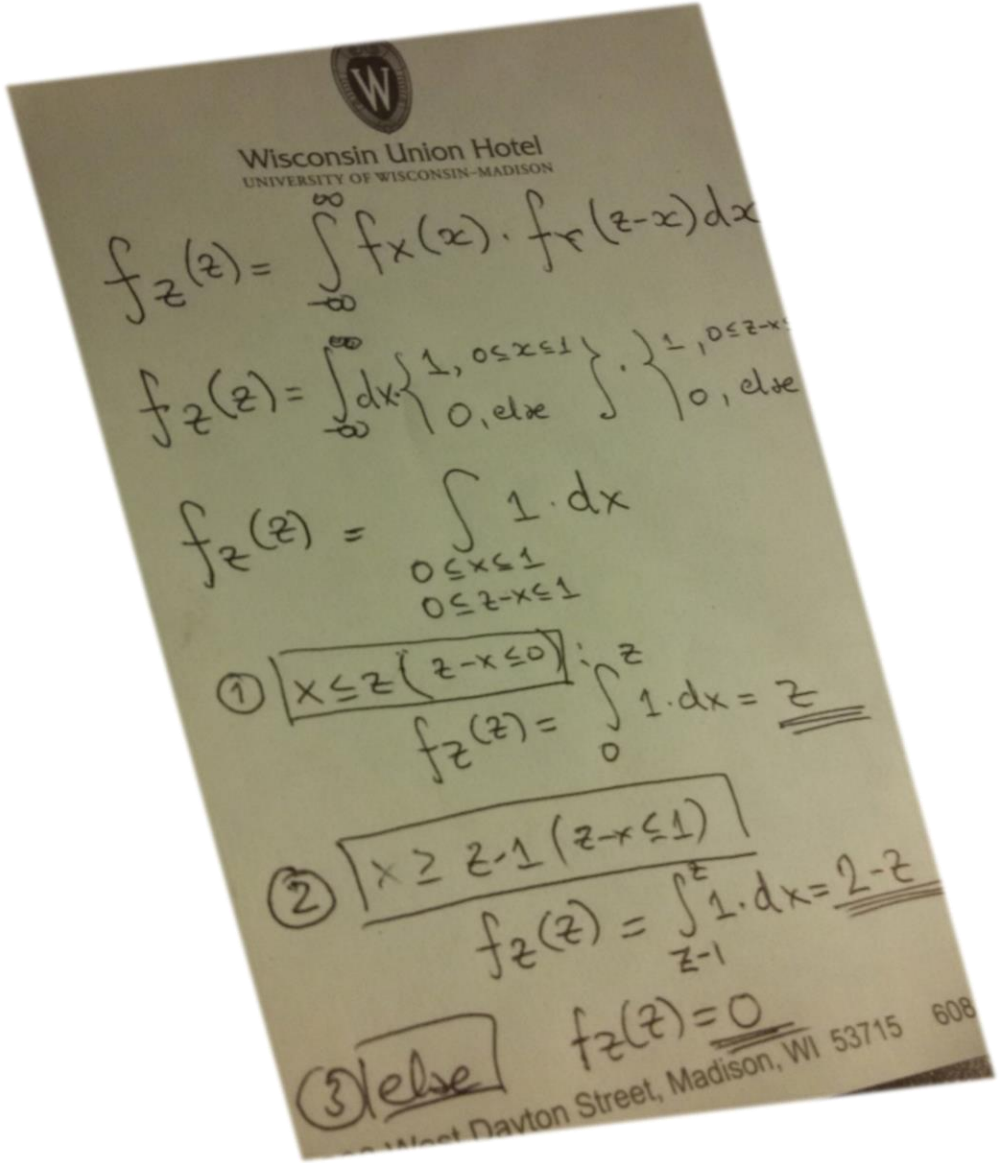
$X \sim \mathcal{U}(0,1)$

$Y \sim \mathcal{U}(0,1)$

X and Y are **i.i.d.**

$Z = X+Y$

What is PDF of Z?



But How About This?

$X \sim \mathcal{U}(0,1)$

$Y \sim \mathcal{U}(0,1)$

X and Y are **i.i.d.**

$Z = X+Y$

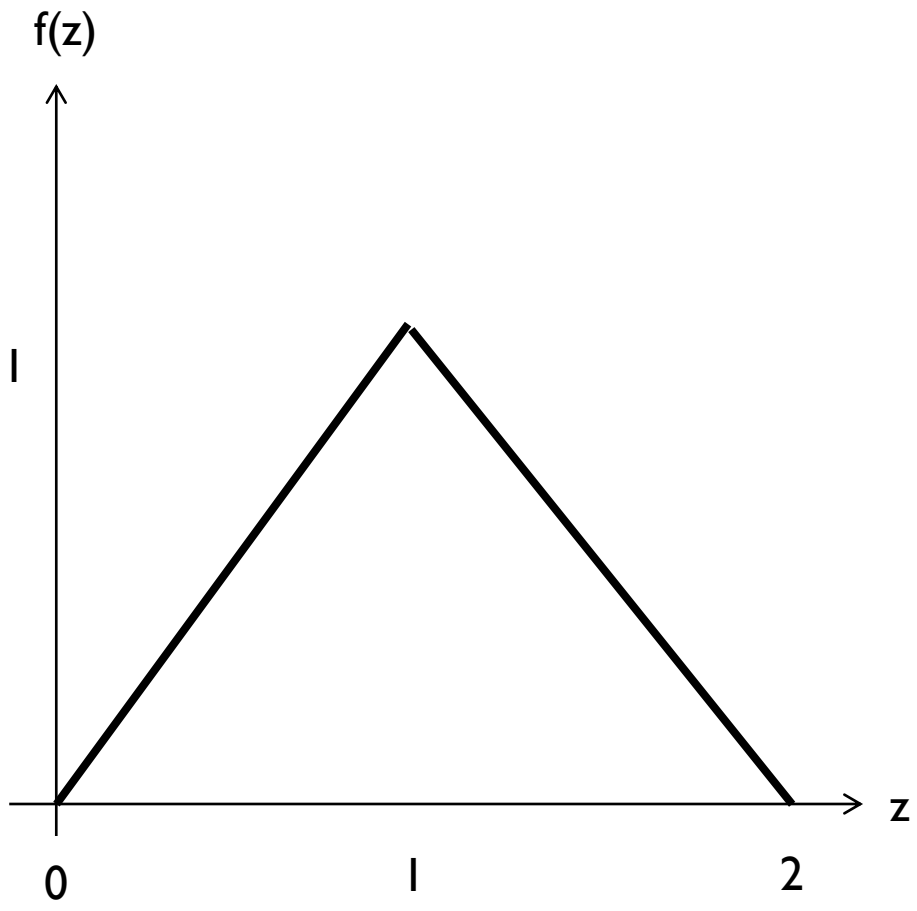
What is PDF of Z?

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y

return Z
```

www.psisolver.org

$ psi sum_uniform.prb

But How About This?

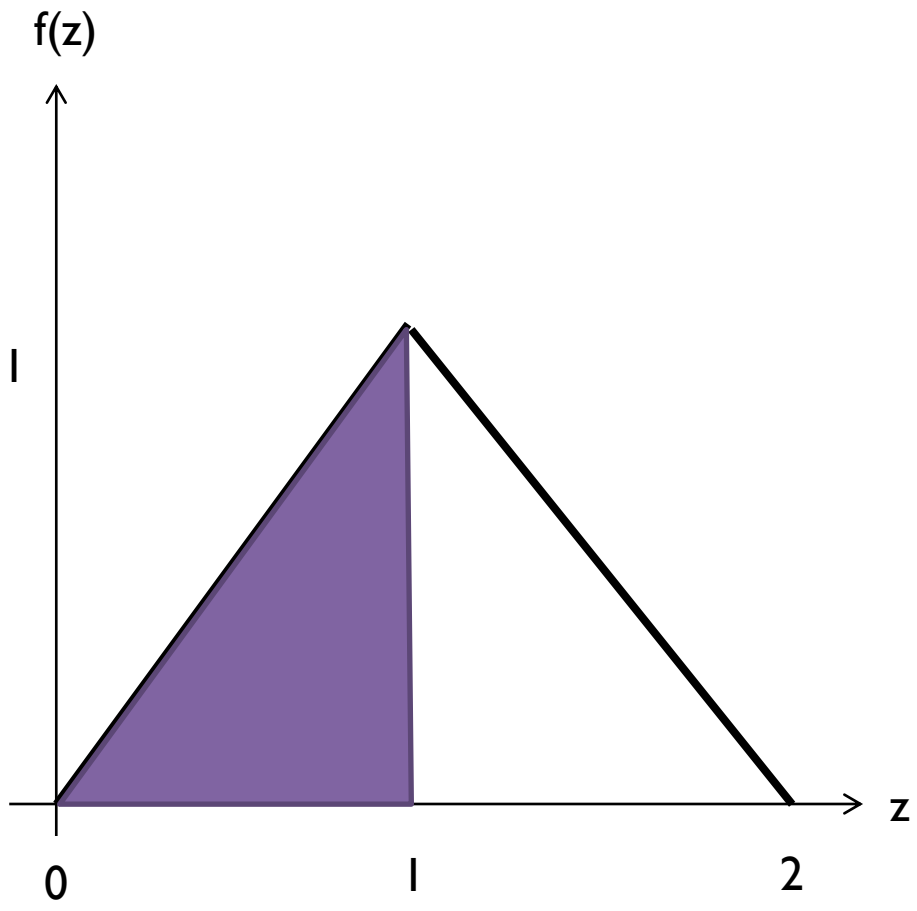


$$\begin{cases} 2 - Z & 1 \leq Z \leq 2 \\ Z & 0 \leq Z < 1 \end{cases}$$

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y

return Z
```

$ psi sum_uniform.prb

But How About This?

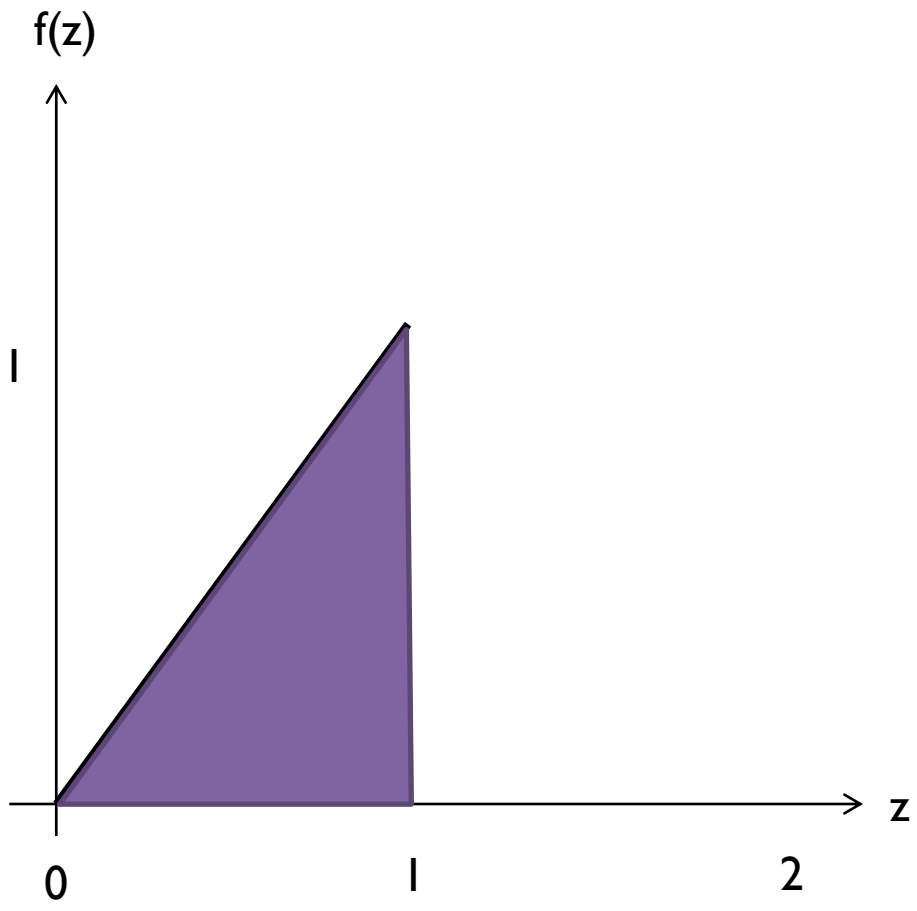


$$\begin{cases} Z & 0 \le Z < 1 \end{cases}$$

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```

$ psi sum_uniform.prb

But How About This?

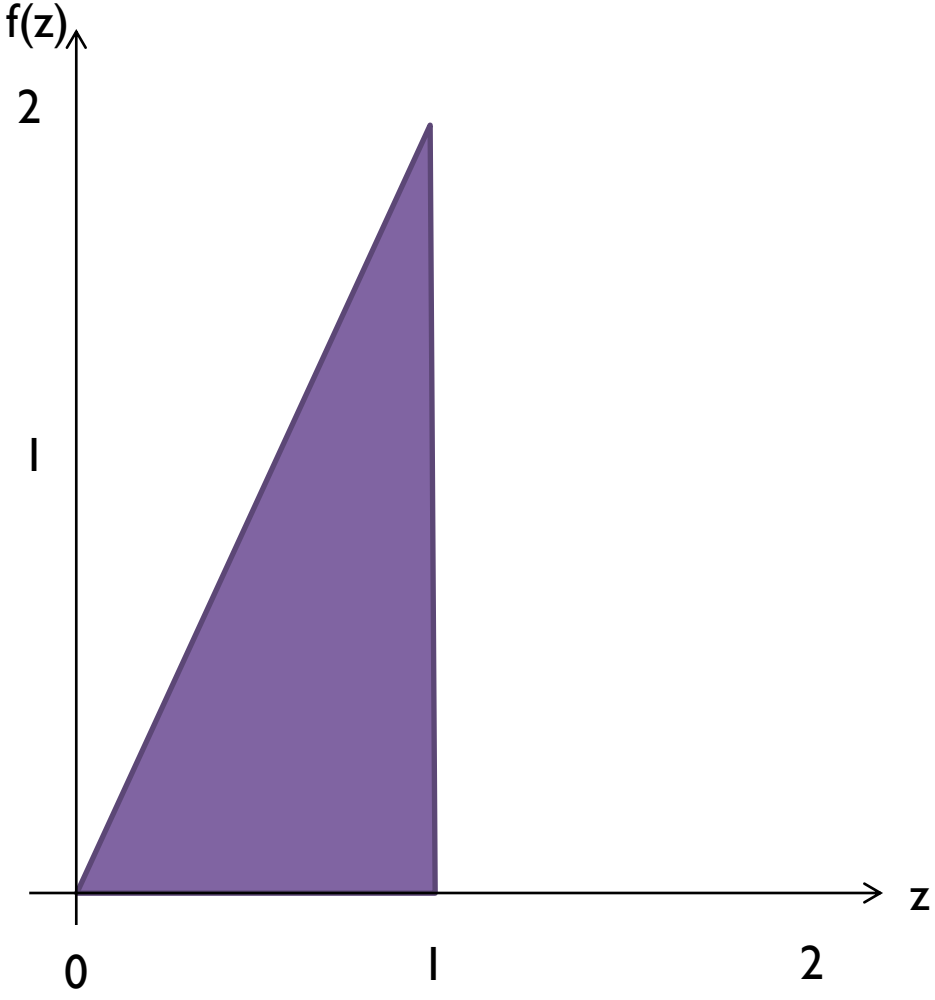


$$\begin{cases} Z & 0 \le Z < 1 \end{cases}$$

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```

$ psi sum_uniform.prb

But How About This?



$$\begin{cases} 2Z & 0 \le Z < 1 \end{cases}$$

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```

$ psi sum_uniform.prb

Probabilistic Programs

Extend Standard (Deterministic) Programs

Distribution	`X := Uniform(0, 1);`
Assertion	`assert ( X >= 0 );`
Observation	`observe ( X >= 0.5 );`
Query	`return X;`

Probabilistic Programs

Extend Standard (Deterministic) Programs

Distribution	`X := Uniform(0, 1);`
Assertion	`assert ( X >= 0 );`
Observation	`observe ( X >= 0.5 );`
Query	`return X;`

Promise of Probabilistic Programming

Decoupling the model from solving

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```

PROBABILISTIC INFERENCE ENGINE

Promise of Probabilistic Programming

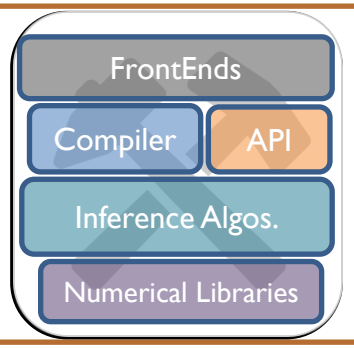
Decoupling the model from solving

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt$$

$$\begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix}$$

$$f(t) * g(t) := \int_{-\infty}^{\infty} f(\tau) g(t-\tau)\, d\tau$$

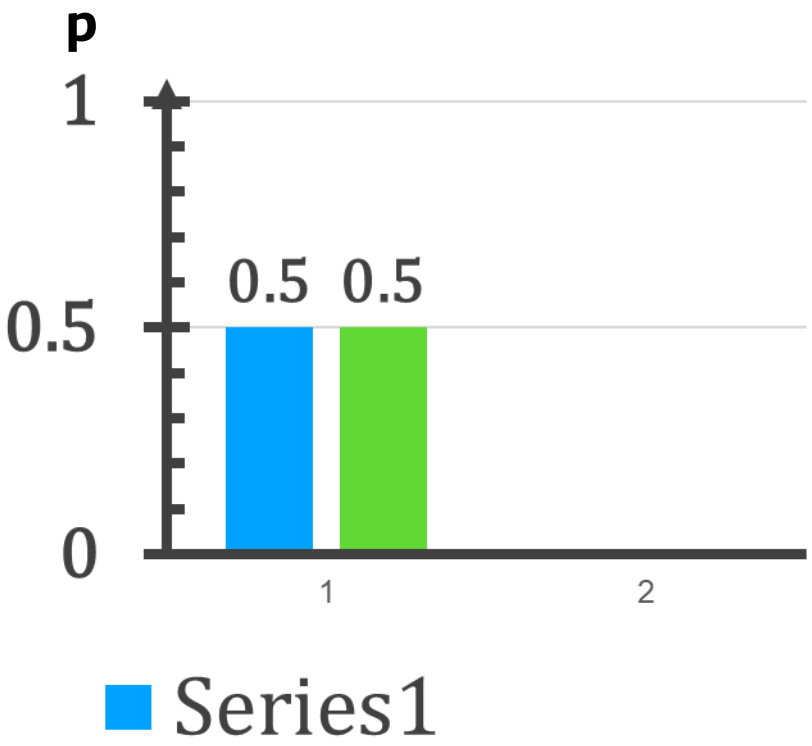


Probabilistic Model

$$A \sim Bernoulli(0.5)$$

$$P(A = 1)$$

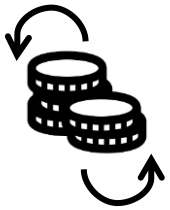
$$head: 1$$
$$tail: \quad 0$$



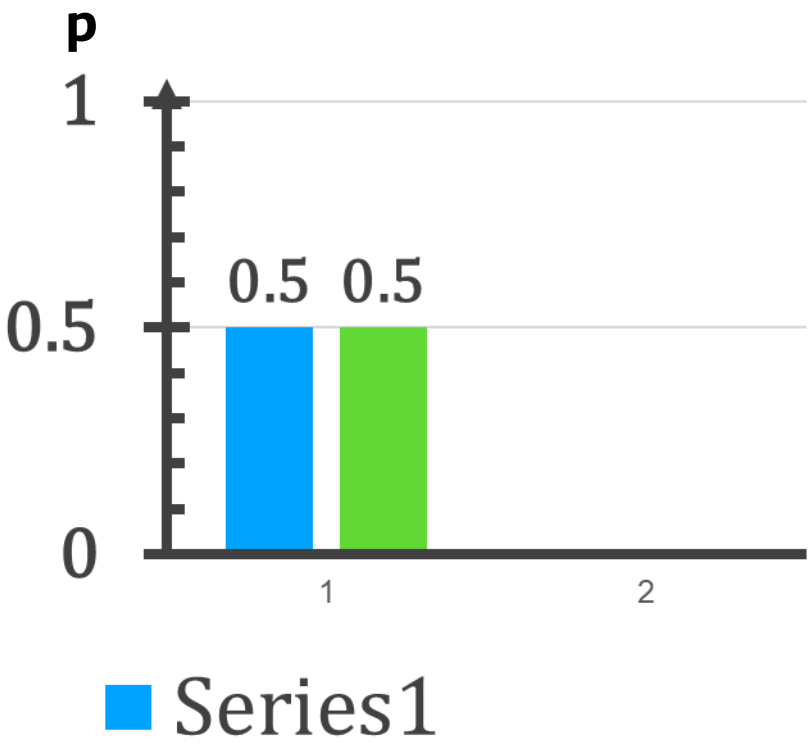
Probabilistic Model

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

$$P(A = 1)$$



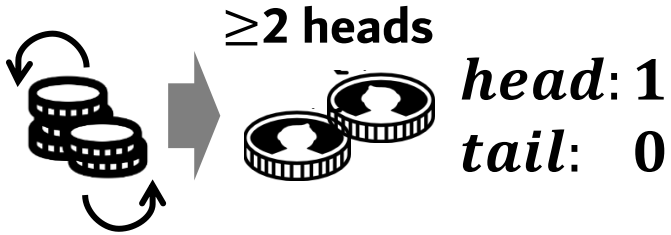
$$head: 1$$
$$tail: \quad 0$$



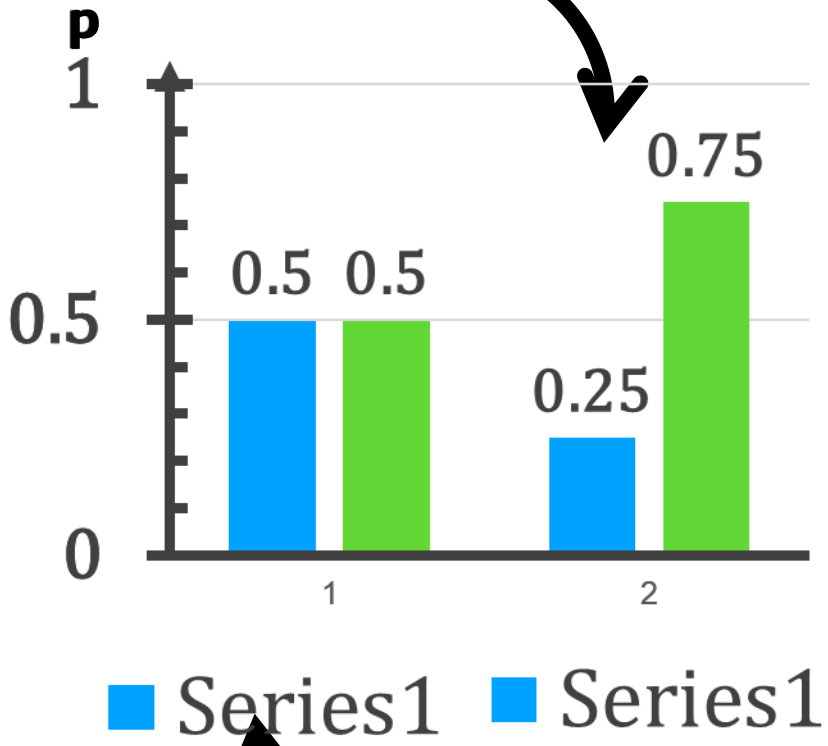
Probabilistic Model

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

$$P(A = 1 | A + B + C \geq 2)$$



Posterior Distribution

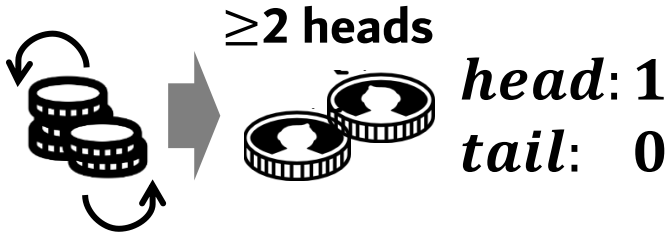


Prior Distribution

Probabilistic Programming

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

$$P(A = 1 | A + B + C \geq 2)$$

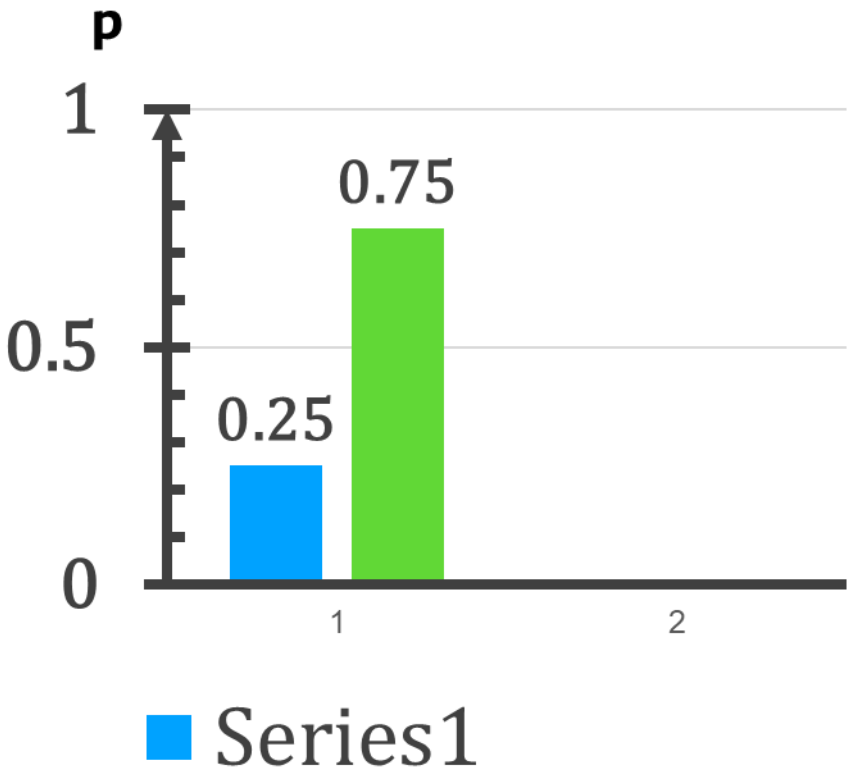



```
function main(){
   A=flip(0.5);
   B=flip(0.5);
   C=flip(0.5);

   condition(A+B+C>=2);
   return A;
}
```

Probabilistic Programming

```
function main(){
   A=flip(0.5);
   B=flip(0.5);
   C=flip(0.5);

   condition(A+B+C>=2);
   return A;
}
```

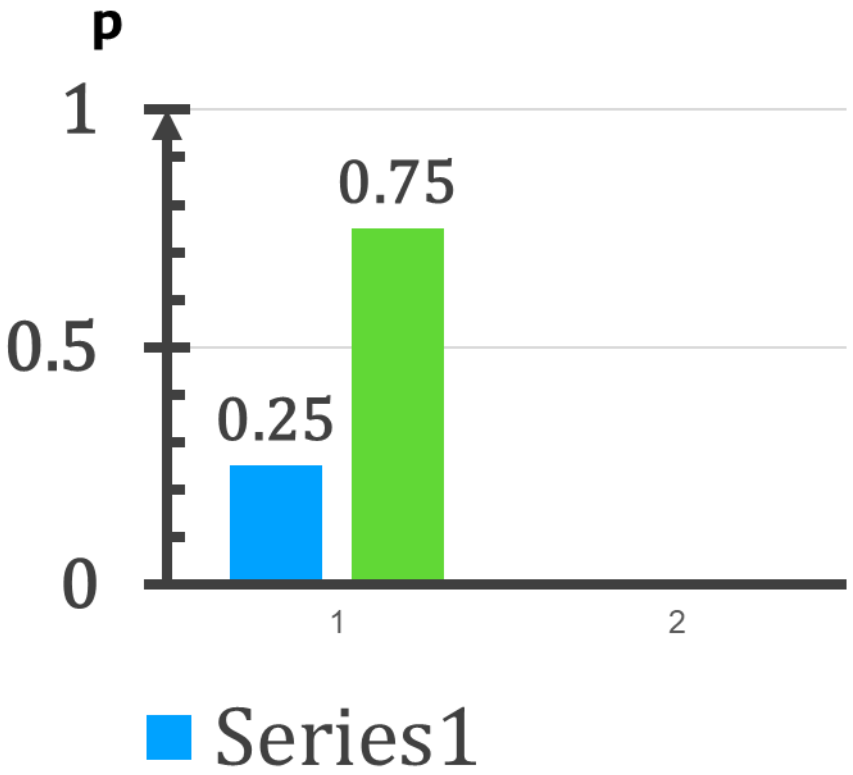
Inference Engine



Probabilistic Programming

```
function main(){
   A=flip(0.5);
   B=flip(0.5);
   C=flip(0.5);

   condition(A+B+C>=2);
   return A;
}
```

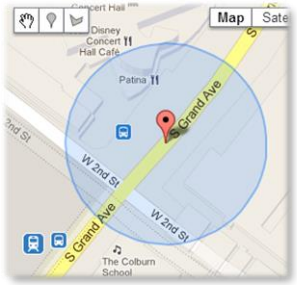
Inference Engine



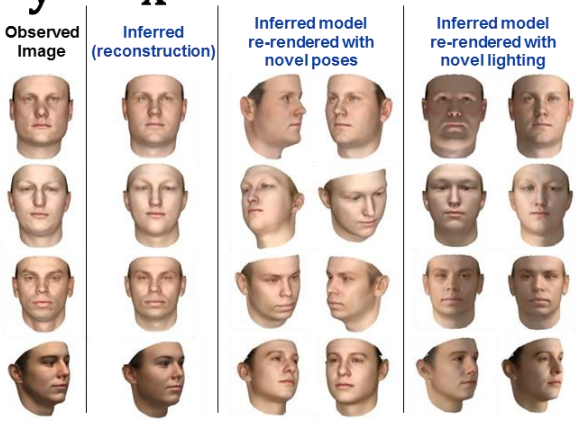
Probabilistic Applications

Modeling of
Complex Systems

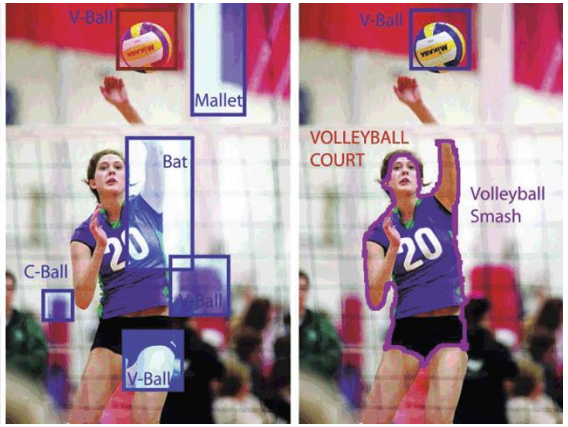
GPS & Navigation



Face Reconstruction



Scene labeling



Spam
Filter

Microsoft

XBOX

facebook

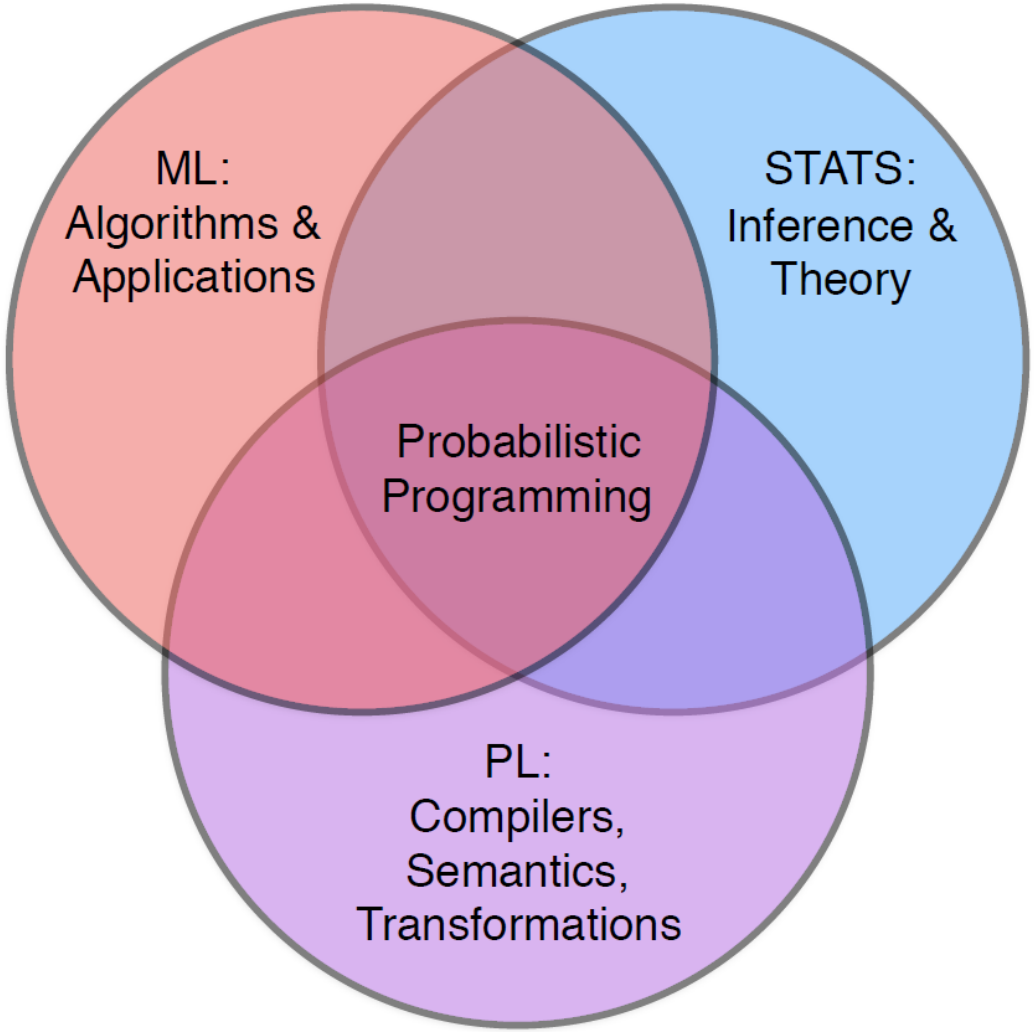
UBER

Pyro

Google

Tensorflow

Probabilistic Programming



Graphic from Frank Wood's lecture

Promise of Probabilistic Programming

Decoupling the model from solving

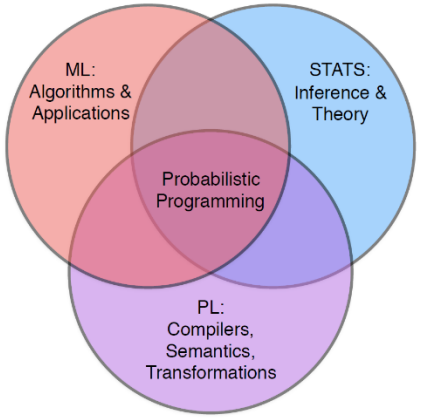
```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```

PROBABILISTIC INFERENCE ENGINE

Promise of Probabilistic Programming

Decoupling the model from solving

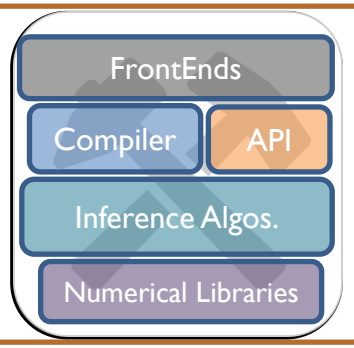
```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```



$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt$$

$$\begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix}$$

$$f(t) * g(t) := \int_{-\infty}^{\infty} f(\tau) g(t-\tau)\, d\tau$$



Declarative Promise of ~~Probabilistic~~ Programming

Decoupling the model from solving

```
from z3 import *
x = Real('x')
y = Real('y')
s = Solver()
s.add(x + y > 5,
      x > 1, y > 1)
return s.check()
```

SMT SOLVERS

```
X := Uniform(0,1)
Y := Uniform(0,1)

Z := X + Y
observe Z < 1
return Z
```

PROBABILISTIC INFERENCE ENGINE

Example Language:

WWW.WEBPPL.ORG

Probability Refresher

2.1. Basic definition.

We define a *probability triple* or *(probability) measure space* or *probability space* to be a triple $(\Omega, \mathcal{F}, \mathbf{P})$, where:

- the *sample space* Ω is any non-empty set (e.g. $\Omega = [0,1]$ for the uniform distribution considered above);
- the *σ-algebra* (read "sigma-algebra") or *σ-field* (read "sigma-field") $\mathcal{F}$ is a collection of subsets of Ω, containing Ω itself and the empty set $\emptyset$, and closed under the formation of complements* and countable unions and countable intersections (e.g. for the uniform distribution considered above, $\mathcal{F}$ would certainly contain all the intervals $[a,b]$, but would contain many more subsets besides);
- the *probability measure* $\mathbf{P}$ is a mapping from $\mathcal{F}$ to $[0,1]$, with $\mathbf{P}(\emptyset) = 0$ and $\mathbf{P}(\Omega) = 1$, such that $\mathbf{P}$ is countably additive as in (1.2.3).

Rosenthal J.; A First Look at Rigorous Probability Theory 2 ed.

Probability Refresher

Probability Distribution

- Discrete Distributions
- Continuous Distributions
- Hybrid Joint Distributions

Discrete Distribution: Bernoulli

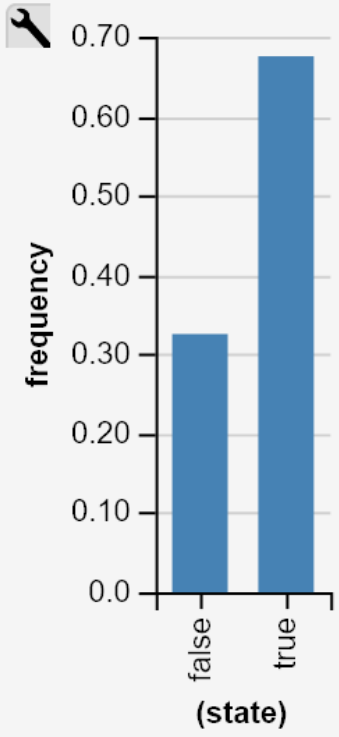
Model

Inference

Visualization

```
var main = function () {
    return bernoulli(0.67)
}

var dist = Infer( {method: 'MCMC', samples: 100000}, main);

viz.auto(dist);
```

run



Probability mass function (PMF)

Discrete Distribution: Uniform

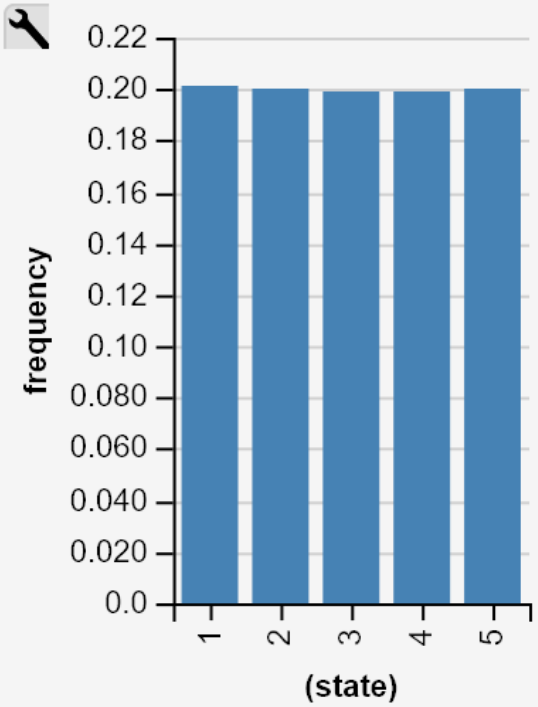
Model

Inference

Visualization

```
var main = function () {
    return categorical( { vs: [1,2,3,4,5] } )
}

var dist = Infer( {method: 'MCMC', samples: 100000}, main);

viz.auto(dist);
```

run



Discrete Distribution: Binomial

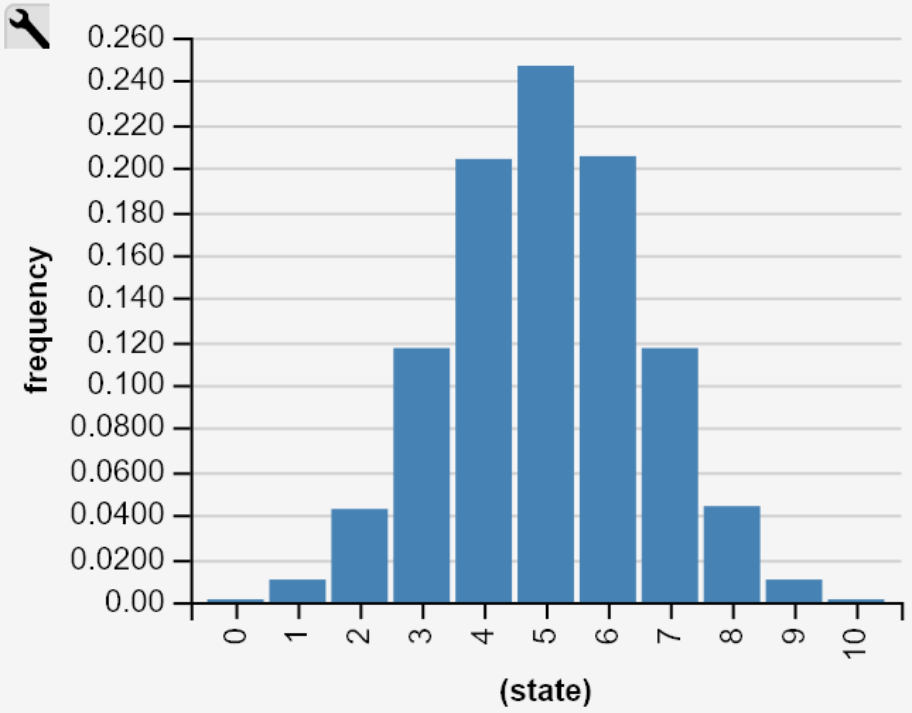
Model

Inference

Visualization

```
var main = function () {
    return binomial(0.5, 10)
}

var dist = Infer( {method: 'MCMC', samples: 100000}, main);

viz.auto(dist);
```

run



Model

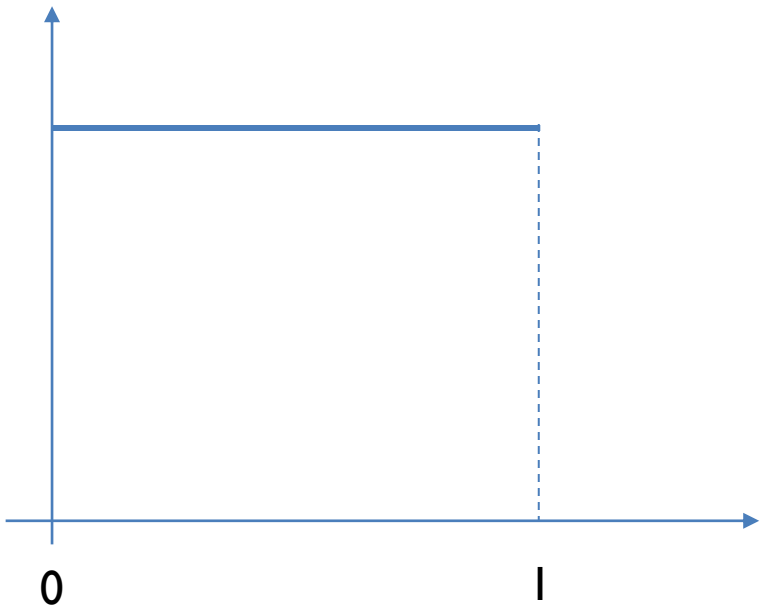
Inference

Visualization

```
var main = function () {
    return uniform(0,1)
}

var dist = Infer({method: 'MCMC', samples: 10000},  main);

viz.auto(dist);
```

run

Continuous Distribution: Uniform



Continuous Distribution: Uniform

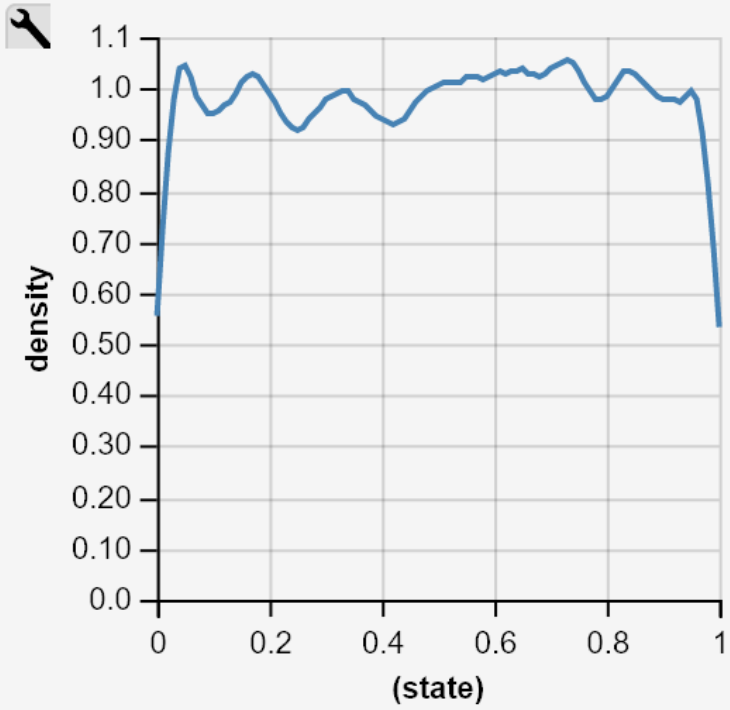
Model

Inference

Visualization

```
var main = function () {
    return uniform(0,1)
}

var dist = Infer({method: 'MCMC', samples: 10000},  main);

viz.auto(dist);
```

run



Continuous Distribution: Gaussian

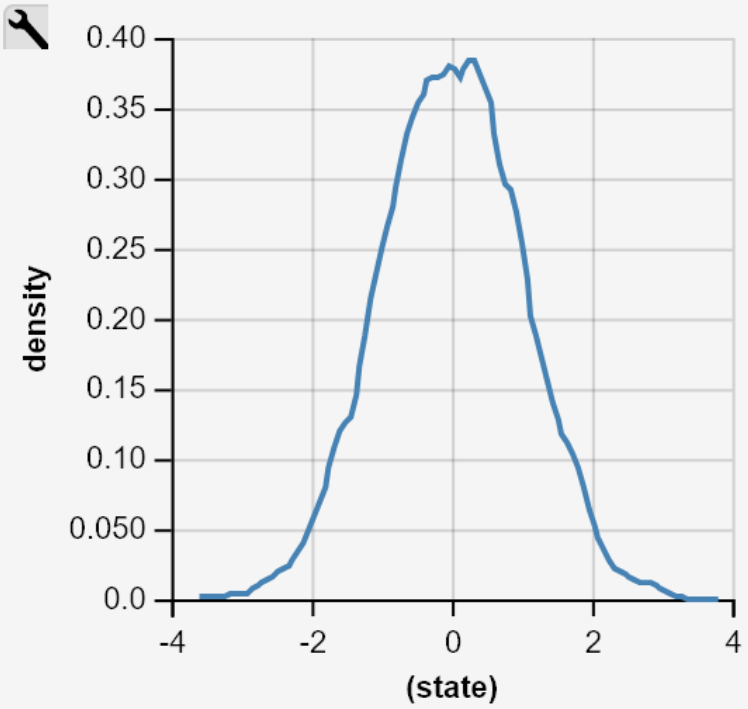
Model

Inference

Visualization

```
var main = function () {
    return gaussian(0,1)
}

var dist = Infer({method: 'MCMC', samples: 10000},  main);

viz.auto(dist);
```

run



Probability density function (PDF)

Distribution Function

Probability Mass Function: $\Pr(X = x)$

Probability Density Function: $f(x)$

For background see e.g:
https://www.cs.ubc.ca/~fwood/CS532W-539W/notes/essentials-of-bayesian-inference.pdf

Distribution Function

Probability Mass Function: $\Pr(X = x)$

Probability Density Function: $f(x)$

Cumulative Distribution Function

- Sum of PMF $\Pr(X \leq x) = \sum_{u=x_{min}}^{x} \Pr(X = u)$
- or the integral of PDF $\Pr(X \leq x) = \int_{x_{min}}^{x} f(u)du$

For background see e.g:
https://www.cs.ubc.ca/~fwood/CS532W-539W/notes/essentials-of-bayesian-inference.pdf

Expectation and Variance

Expected value: measure of central tendency

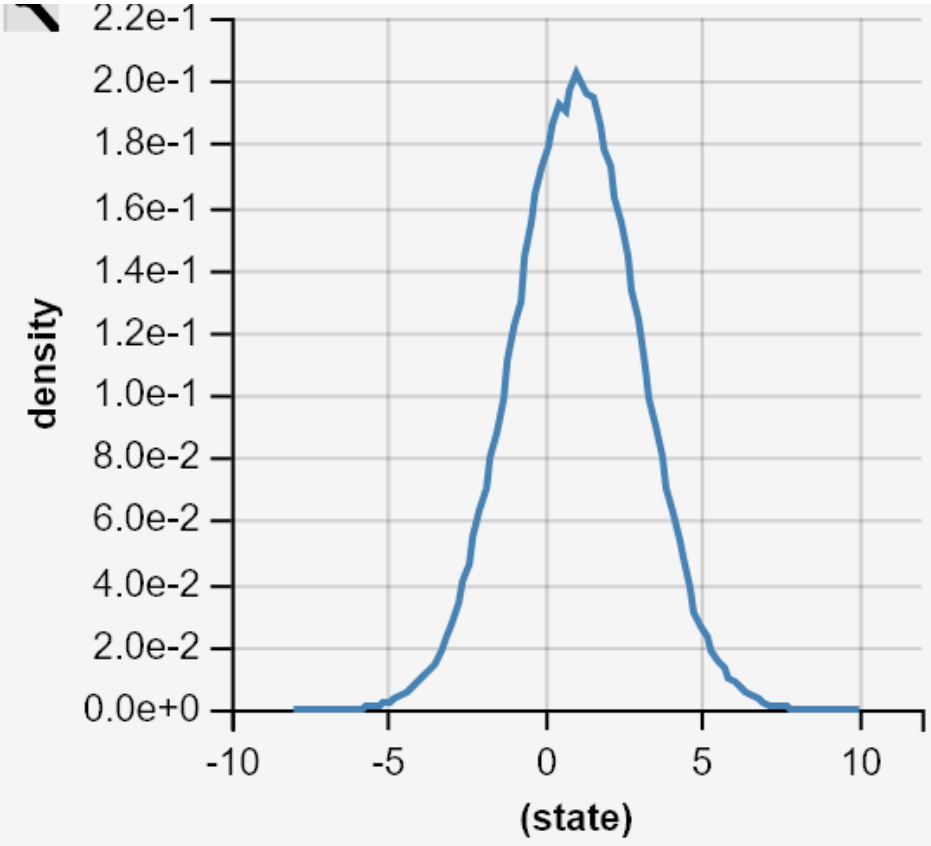
$$E(X) = \sum x \cdot P_X(x) \quad \text{or} \quad E(X) = \int x \cdot f_X(x)\, dx$$

Variance: measure of spread

$$Var(X) = E(X^2) - (EX)^2$$

Mode: value with highest probability from the posterior

```
var main = function () {
    return gaussian( 1, 2 )
}

var dist = Infer( {method: 'MCMC',
                   samples: 100000}, main);

viz.auto(dist);

// Expectation
var e = expectation(dist)
// Variance
var v = expectation(dist, function(x) {x^2}) - e^2

print( "Expectation: " + e + "\nVariance: " + v)
```




```
Expectation: 0.9967447453851697
Variance: 2
```

Reminder: Independence

Definition:

$$\boldsymbol{Pr(X, Y) = Pr(X) \cdot Pr(Y)}$$

But also*:

$$\boldsymbol{Pr(X \mid Y) = Pr(X)}$$
$$\boldsymbol{Pr(Y \mid X) = Pr(Y)}$$

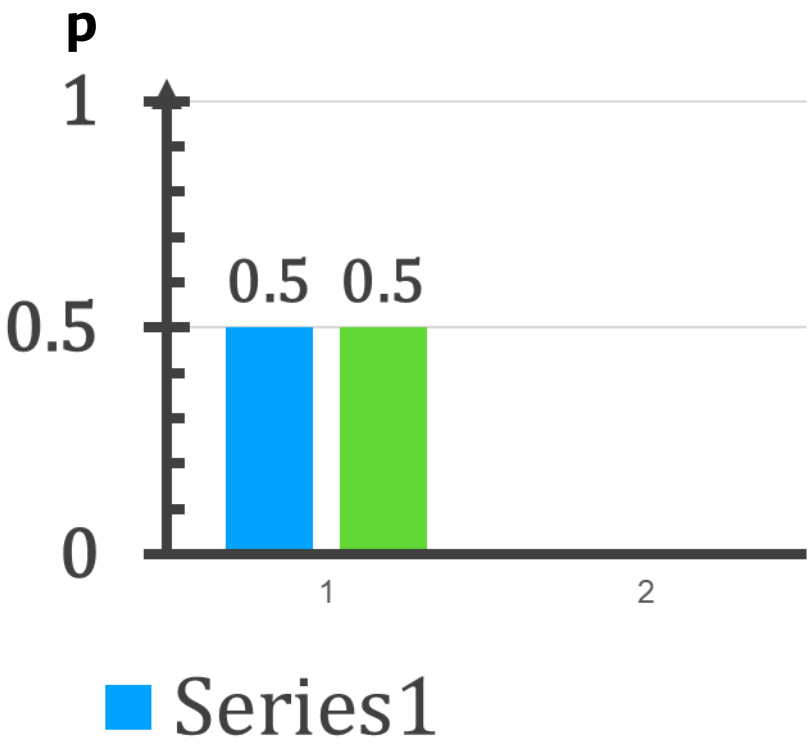
*Using the fact that for any two variables $\boldsymbol{Pr(X, Y) = Pr(X|Y) \cdot Pr(Y)}$

Reminder

$$A \sim Bernoulli(0.5)$$

$$P(A = 1)$$

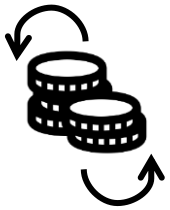
$$head: 1$$
$$tail: \ 0$$



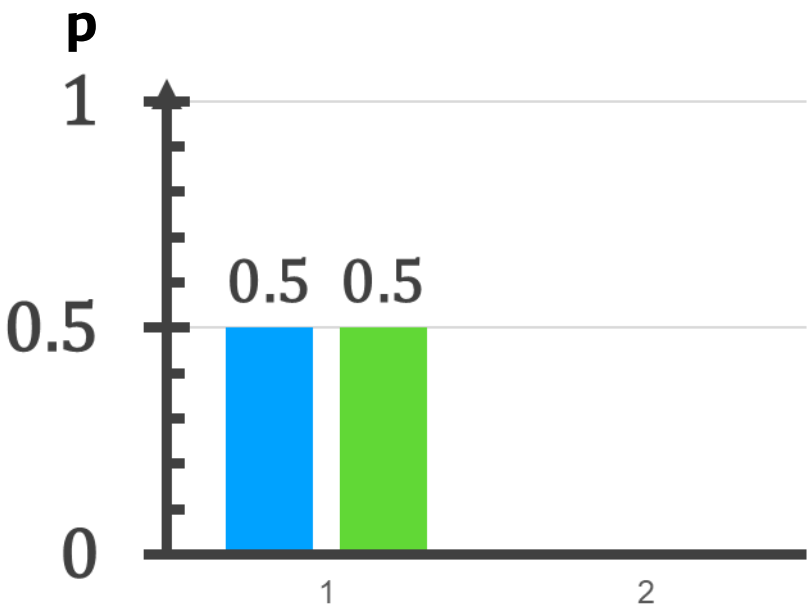
Reminder

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

$$P(A = 1)$$



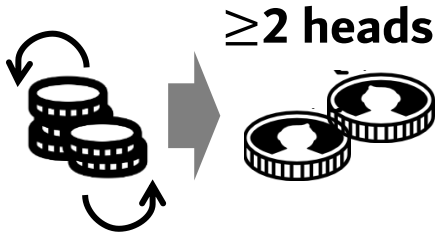
$$head: 1$$
$$tail: \quad 0$$



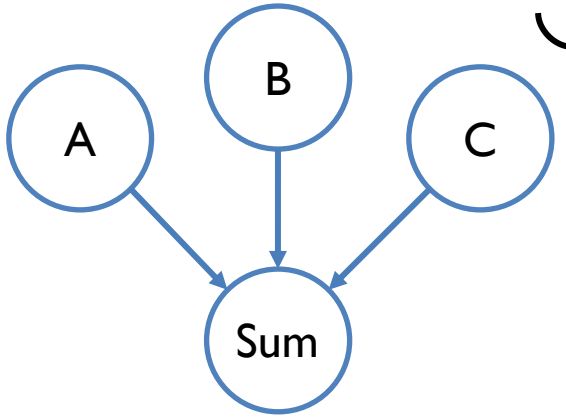
Reminder: Observations can induce dependence

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

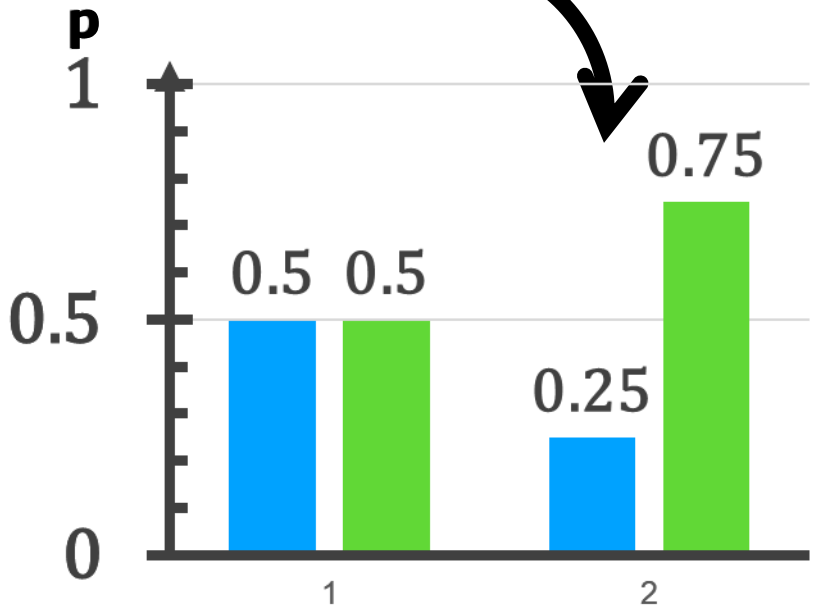
$$P(A = 1 | A + B + C \geq 2)$$



$head$: 1
$tail$: 0



Posterior Distribution

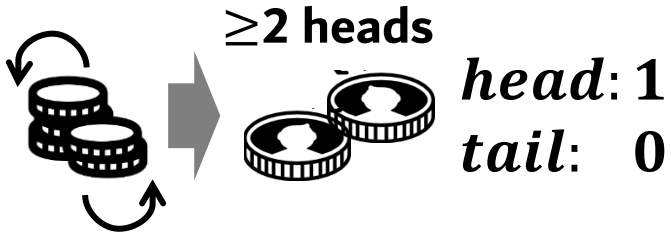


Prior Distribution

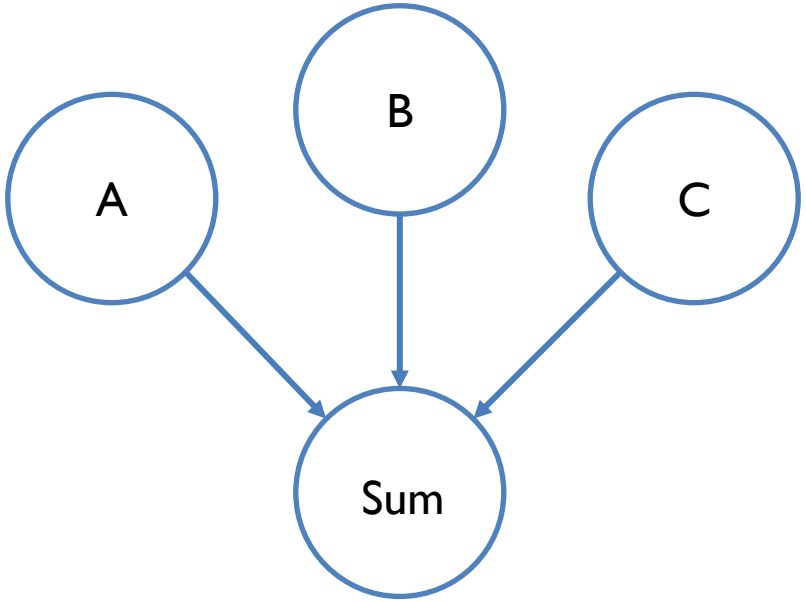
Reminder: Observations can induce dependence

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

$$P(A = 1 | A + B + C \geq 2)$$



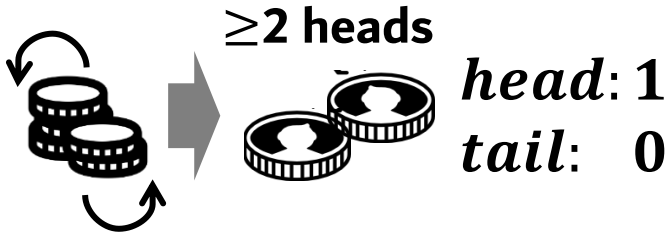
Dependence Structure



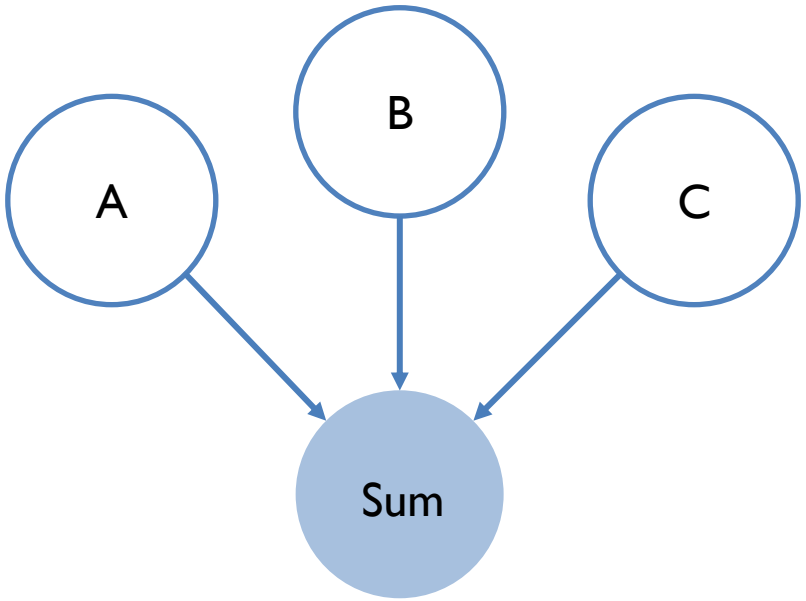
Reminder: Observations can induce dependence

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

$$P(A = 1 | A + B + C \geq 2)$$



Dependence Structure



Search for more: Bayesian Networks

Bayes' Rule

Belief Revision

Thomas Bayes*
1701 –1761

Bayes' Rule

Belief Revision

Hypothesis

$$\Pr(\theta \mid x) = \frac{\Pr(x \mid \theta) \cdot \Pr(\theta)}{\Pr(x)}$$

Data

Bayes' Rule

Belief Revision

Hypothesis

$\Theta \in \{\theta_1, \theta_2, \ldots, \theta_n\}$

$$\Pr_{\Theta|X}(\Theta = \theta \mid X = x) = \frac{\Pr_{X|\Theta}(X = x \mid \Theta = \theta) \cdot \Pr_{\Theta}(\Theta = \theta)}{\Pr_X(X = x)}$$

Data: $X \in \{x_1, x_2, \ldots, x_n\}$

Full form: Each PMF/PDF is a different function,

Each random variable can have values from its support set

Bayes' Rule

Belief Revision

Likelihood

Prior Distribution

Posterior Distribution

$$\Pr(\theta \mid x) = \frac{\Pr(x \mid \theta) \cdot \Pr(\theta)}{\Pr(x)}$$

Normalization Constant

Bayes' Rule

Belief Revision

Posterior Distribution

Likelihood

Prior Distribution

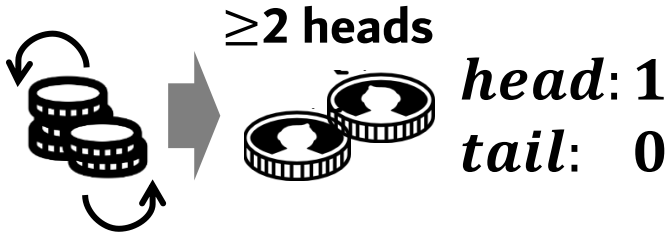
$$\Pr(\theta \mid x) \sim \Pr(x \mid \theta) \cdot \Pr(\theta)$$

Enough to order different interpretations and select the most likely one

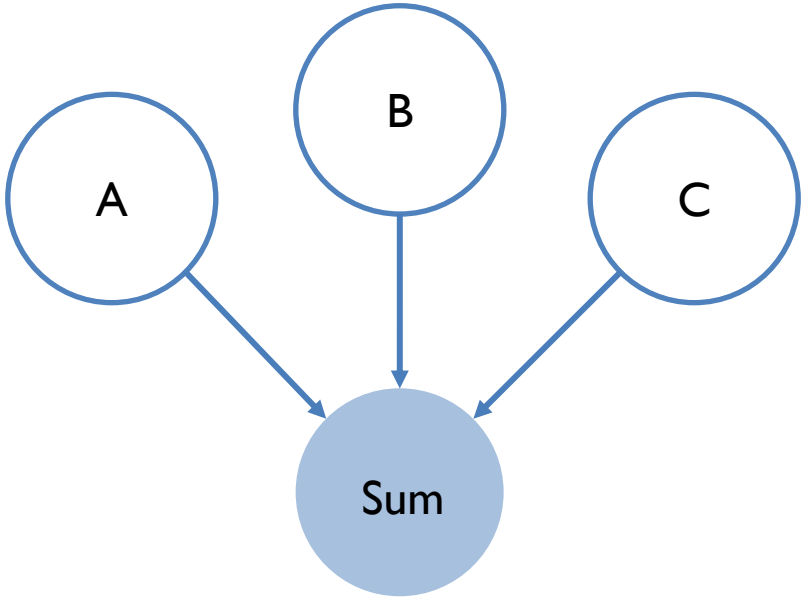
Condition / Observe / Factor

$$A \sim Bernoulli(0.5)$$
$$B \sim Bernoulli(0.5)$$
$$C \sim Bernoulli(0.5)$$

$$P(A = 1 | A + B + C \geq 2)$$

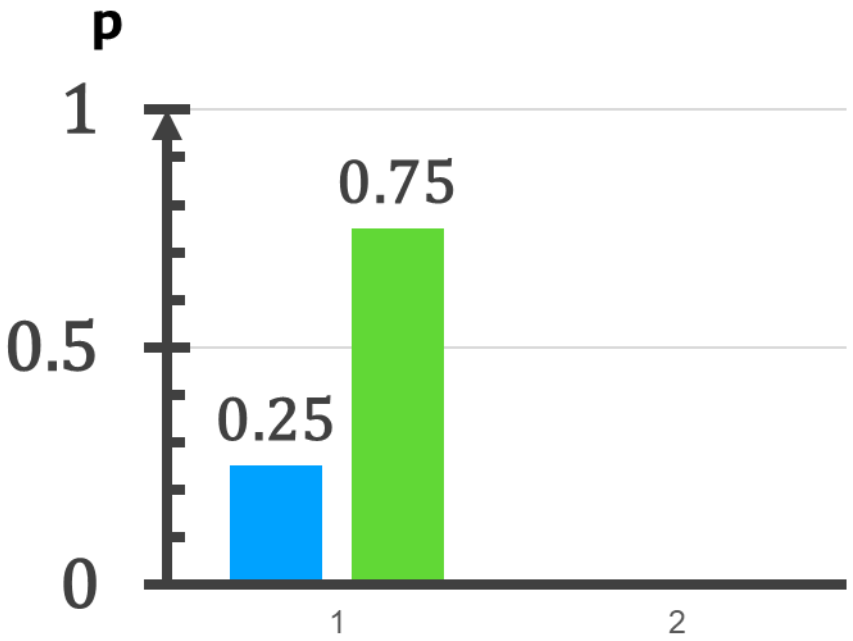


Dependence Structure



Conditioning on Event

```
function main(){
    var A=flip(0.5);
    var B=flip(0.5);
    var C=flip(0.5);

    var Sum=A+B+C
    condition(Sum>=2);
    return A;
}
```



Conditioning on Event: Mechanics

```
function main(){
   var A=flip(0.5);
   var B=flip(0.5);
   var C=flip(0.5);

   var Sum=A+B+C
   condition(Sum>=2);
   return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Conditioning on Event: Mechanics

```
function main(){
   var A=flip(0.5);
   var B=flip(0.5);
   var C=flip(0.5);

   var Sum=A+B+C
   condition(Sum>=2);
   return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0	0	1/8
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Conditioning on Event: Mechanics

```
function main(){
   var A=flip(0.5);
   var B=flip(0.5);
   var C=flip(0.5);

   var Sum=A+B+C
   condition(Sum>=2);
   return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0	0	1/8
0	0	1	1	1/8
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Conditioning on Event: Mechanics

```
function main(){
    var A=flip(0.5);
    var B=flip(0.5);
    var C=flip(0.5);

    var Sum=A+B+C
    condition(Sum>=2);
    return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0	0	1/8
0	0	1	1	1/8
0	1	0	1	1/8
0	1	1	2	1/8
1	0	0	1	1/8
1	0	1	2	1/8
1	1	0	2	1/8
1	1	1	3	1/8

Bayes Formula:

$$\Pr(\theta \mid x) = \frac{\Pr(x \mid \theta) \cdot \Pr(\theta)}{\Pr(x)}$$

Conditioning on Event: Mechanics

```
function main(){
   var A=flip(0.5);
   var B=flip(0.5);
   var C=flip(0.5);

   var Sum=A+B+C
   condition(Sum>=2);
   return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0	0	1/8
0	0	1	1	1/8
0	1	0	1	1/8
0	1	1	2	1/8
1	0	0	1	1/8
1	0	1	2	1/8
1	1	0	2	1/8
1	1	1	3	1/8

$$\Pr(A \mid Sum \geq 2) = \frac{\Pr(Sum \geq 2 \mid A) \cdot \Pr(A)}{\Pr(Sum \geq 2)}$$

Conditioning on Event: Mechanics

```
function main(){
   var A=flip(0.5);
   var B=flip(0.5);
   var C=flip(0.5);

   var Sum=A+B+C
   condition(Sum>=2);
   return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0	0	1/8
0	0	1	1	1/8
0	1	0	1	1/8
0	1	1	2	1/8
1	0	0	1	1/8
1	0	1	2	1/8
1	1	0	2	1/8
1	1	1	3	1/8

$$\Pr(A = 0) = \Pr(A = 1) = 4/8$$

$$\Pr(A \mid Sum \geq 2) = \frac{\Pr(Sum \geq 2 \mid A) \cdot \Pr(A)}{\Pr(Sum \geq 2)}$$

Conditioning on Event: Mechanics

```
function main(){
    var A=flip(0.5);
    var B=flip(0.5);
    var C=flip(0.5);

    var Sum=A+B+C
    condition(Sum>=2);
    return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0	0	1/8
0	0	1	1	1/8
0	1	0	1	1/8
0	1	1	2	1/8
1	0	0	1	1/8
1	0	1	2	1/8
1	1	0	2	1/8
1	1	1	3	1/8

$$\Pr(A = 0) = \Pr(A = 1) = 4/8$$

$$\Pr(Sum \geq 2) = 4/8$$

$$\Pr(A \mid Sum \geq 2) = \frac{\Pr(Sum \geq 2 \mid A) \cdot \Pr(A)}{\Pr(Sum \geq 2)}$$

Conditioning on Event

```
function main(){
    var A=flip(0.5);
    var B=flip(0.5);
    var C=flip(0.5);

    var Sum=A+B+C
    condition(Sum>=2);
    return A;
}
```

A	B	C	Sum	Pr(A,B,C,Sum)
0	0	0	0	1/8
0	0	1	1	1/8
0	1	0	1	1/8
0	1	1	2	1/8
1	0	0	1	1/8
1	0	1	2	1/8
1	1	0	2	1/8
1	1	1	3	1/8

$$\Pr(A \mid Sum \geq 2) = \frac{\Pr(Sum \geq 2 \mid A) \cdot \Pr(A)}{\Pr(Sum \geq 2)}$$

$$\Pr(A = 0) = \Pr(A = 1) = 4/8$$

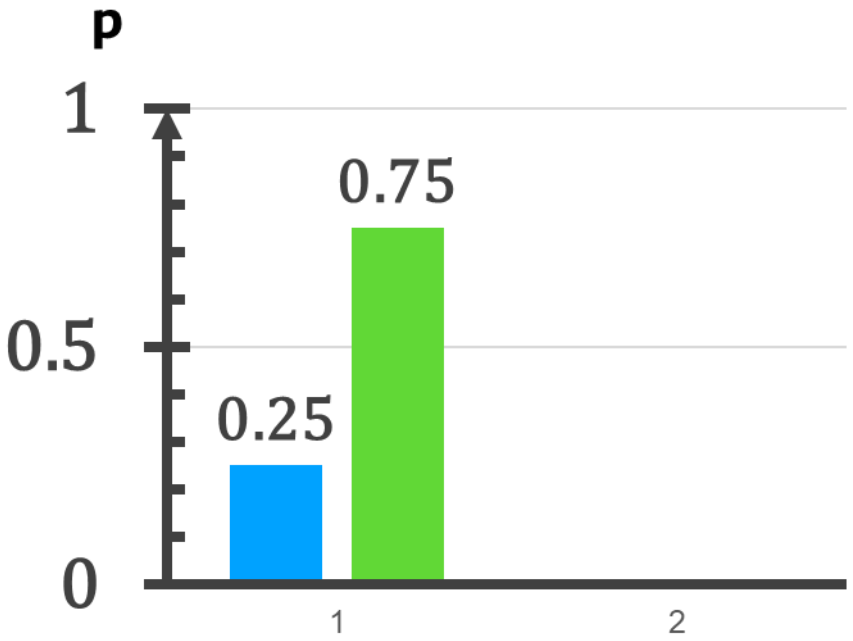
$$\Pr(Sum \geq 2) = 4/8$$

$$\Pr(Sum \geq 2 \mid A = 1) = 3/8$$

$$\Pr(Sum \geq 2 \mid A = 0) = 1/8$$

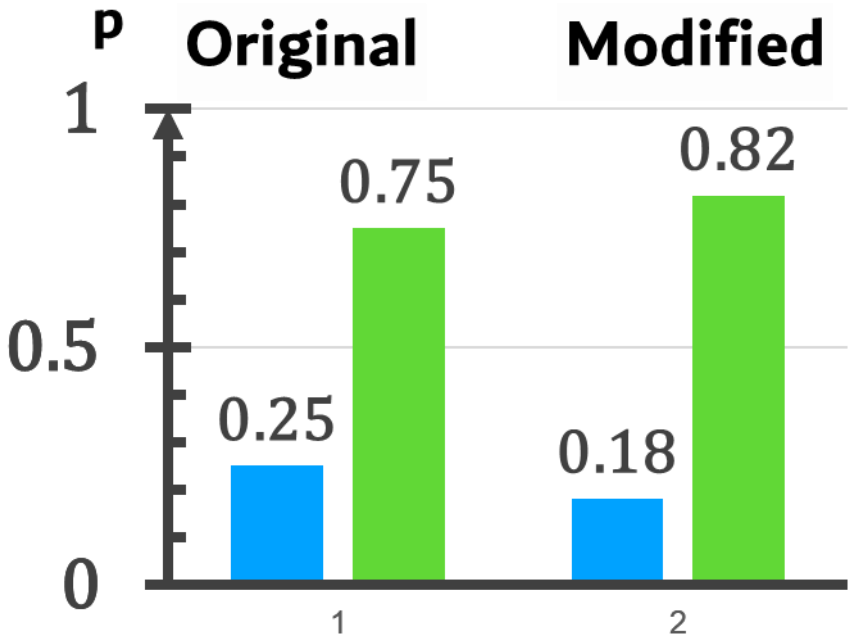
Conditioning on Event

```
function main(){
    var A=flip(0.5);
    var B=flip(0.5);
    var C=flip(0.5);

    var Sum=A+B+C
    condition(Sum>=2);
    return A;
}
```



Conditioning on Event

```
function main(){
    var A=flip(0.5+0.1);
    var B=flip(0.5);
    var C=flip(0.5);

    condition(A+B+C>=2);
    return A;
}
```



When parameter changes,
the change in the posterior is non-linear

A Small Practical Problem

- A system consists of three components that work in parallel
- Each component has the same probability of failure pf=0.1
- The monitoring system compares the results of the three components and finds that the result is correct (i.e., that at least two components produced that result)
- What is the probability that Component#1 was reliable?

A Solution

```
main = function (){
    var pf=0.1
    var A=flip(1-pf);
    var B=flip(1-pf);
    var C=flip(1-pf);

    var Rel=A+B+C
    condition(Rel>=2);
    return A;
}
```

```
var dist = Infer(
{method: 'enumerate'}, main);
viz.table(dist);
```

Pr(A,B,C | Rel >=2)

0	1	2	3	probability
true	true	true	3	0.75
false	true	true	2	0.08333333333333329
true	false	true	2	0.08333333333333329
true	true	false	2	0.08333333333333329

Pr(A | Rel >=2)

(state)	probability
true	0.9166666666666667
false	0.08333333333333333

Continuous Models: TrueSkill

TrueSkill:

- Measure player skills in various sports

Each player has an unknown parameter skill that cannot be directly measured (i.e., it is hidden)

What we can observe is how the in-game performance of the player (which depends on the skill) compares to the performance of the other player

TrueSkill Model

Player Skill: Initially, we assume all players have a similar (randomly assigned) skill, centered around starting point

$$Skill_{Player} \sim Gaussian(100, 10)$$

Player Performance: it is based primarily on skill, but in every game it can be higher or lower, based on the player's inspiration

$$Performance_{Player} \sim Gaussian(Skill, 15)$$

Tournament Scores: Each player plays against others, we record the victory if the player's performance was higher than their opponent's

$$Performance_{Player\#1} > Performance_{Player\#2}$$

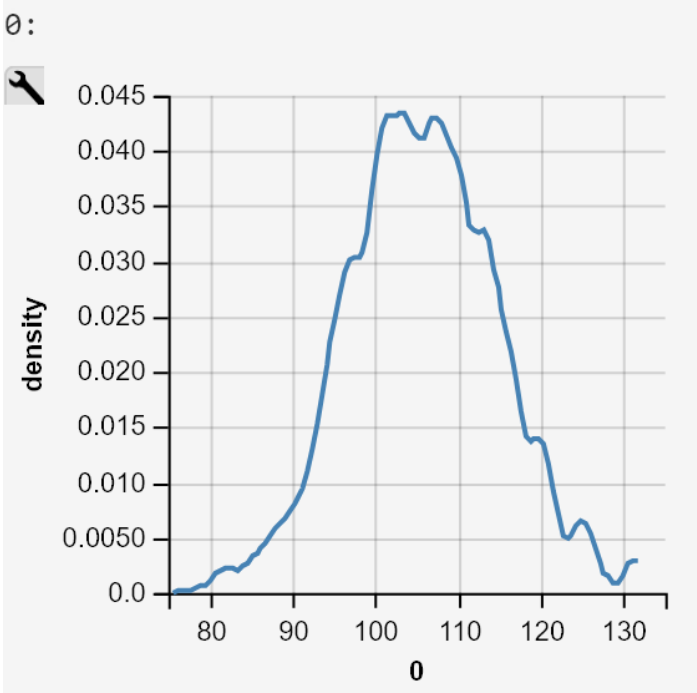
TrueSkill Example: 3 Players

```
var trueSkill = function(){

  var skillA = gaussian(100, 10);
  var skillB = gaussian(100, 10);
  var skillC = gaussian(100, 10);

  var perfA1 = gaussian(skillA, 15), perfB1 = gaussian(skillB, 15);
  condition (perfA1 > perfB1);

  var perfB2 = gaussian(skillB, 15), perfC2 = gaussian(skillC, 15);
  condition (perfB2 > perfC2);

  var perfA3 = gaussian(skillA, 15), perfC3 = gaussian(skillC, 15);
  condition (perfA3 > perfC3);

  return skillA;
}
```

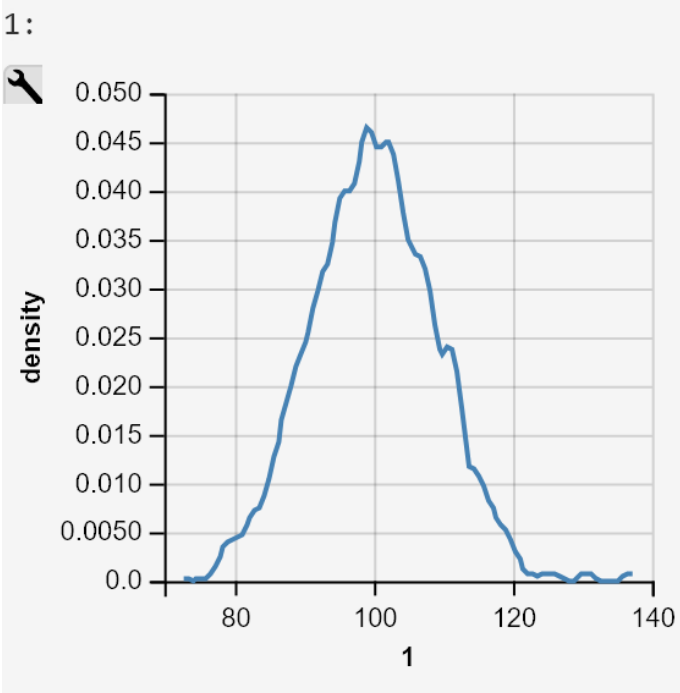
```
var res = Infer({method: 'MCMC', samples:
                 50000}, trueSkill)
print("Expected value: "+expectation(res));
viz.auto(res);
```

Performances

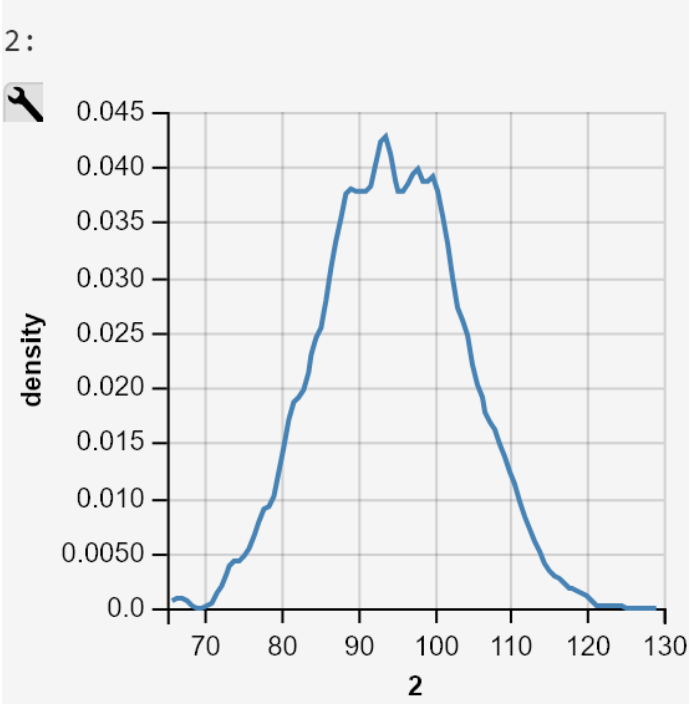
Player A



Player B



Player C



Mean: **105.3** **100.4** **94.6**

Try at Home

Extend the problem to n games:

- The results array contains playes and whether 1st won e.g.,

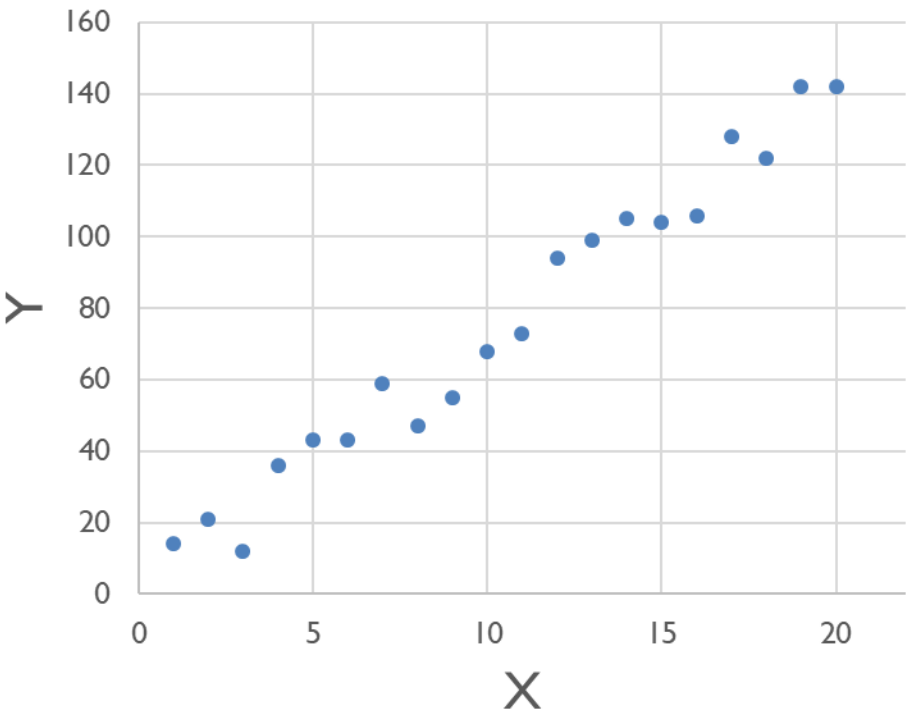
  Res = [['A', 'B', true], ['A', 'B', false], ...]
- Hint: define map M = {'A': skillA, 'B': skillB, 'C': skillC}
- Performance is now indexed by M[Res[0][0]] and M[Res[0][1]]
- Use map operator to condition on all elements of Res

Extend the problem to m players

- Use map operator to generate skills (may use number instead of letter for the player's name)

Continuous Models: Linear Regression

Given a set of points, find a linear relationship that most accurately describes this set



$$Y = \boldsymbol{w} \cdot X + \boldsymbol{b}$$

Slope

Intercept

Linear Regression

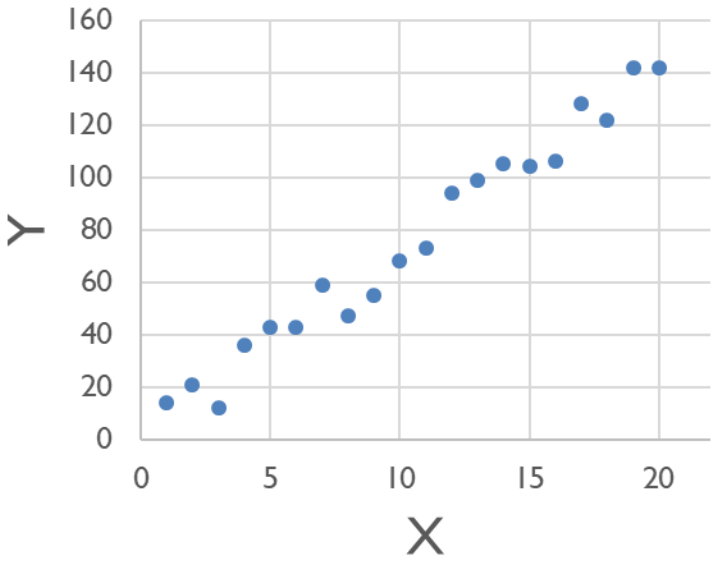
```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Linear Regression Model

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression

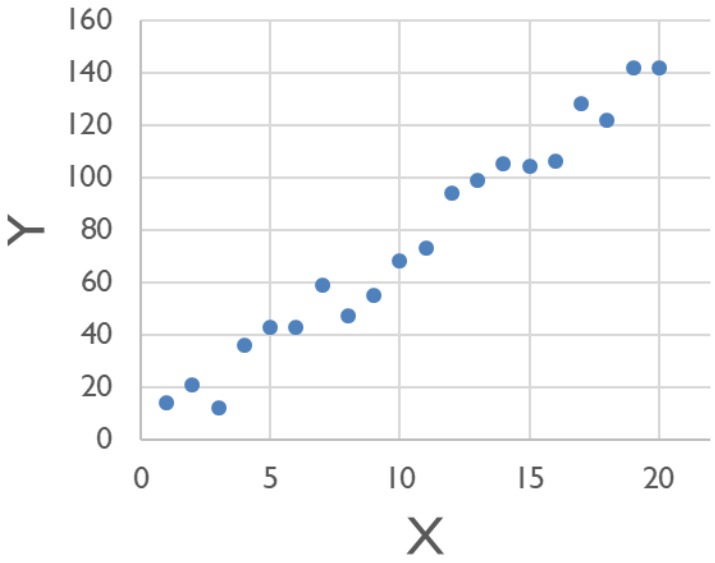
```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Linear Regression

```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

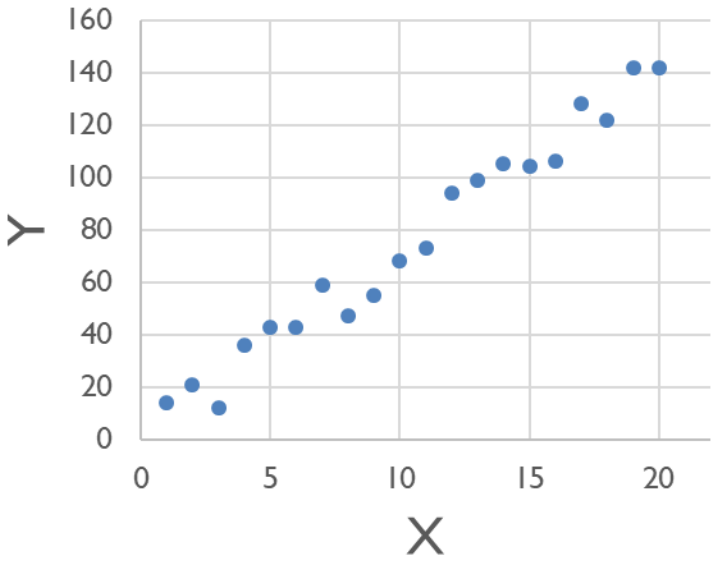
Datasets

```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

Conditioning on Data

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Linear Regression

```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

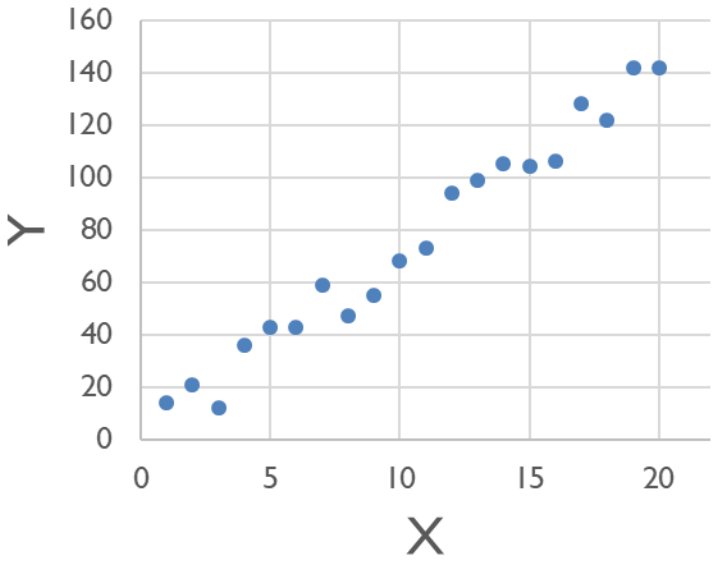
```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

Conditioning on Data

Queries

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Linear Regression

```
x : [1.0, 2.0, ... ];
y : [7.01, 14.2, .... ];
```

Datasets

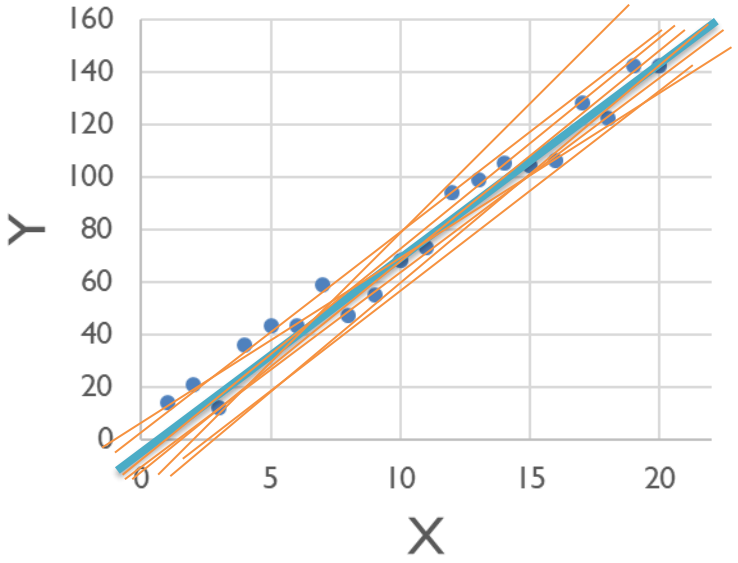
```
w ~ Normal(6 , 10);
b ~ Normal(1 , 5);
observe(y==Normal(w*x + b, 1.0));
posterior w;
posterior b;
```

Priors

Conditioning on Data

Queries

$$Y = \boldsymbol{w}.X + \boldsymbol{b}$$



Linear Regression Model

Continuous Models: Linear Regression

```
var xs = [0, 1, 2, 3];  var ys = [0, 1, 4, 6];

var model = function() {
  var slope = gaussian(0, 2);
  var intercept = gaussian(0, 2);
  var sigma = 1; // for more interesting result, change to gamma(1, 1);

  var f = function(x) { return slope * x + intercept; };

  map2(
    function(x, y) {  observe(Gaussian({mu: f(x), sigma: sigma}), y);  }, xs, ys);
 // function(x, y) {  factor(Gaussian({mu: f(x), sigma: sigma}).score(y));  }, xs, ys);

  return [slope,intercept];
}

viz.marginals({method: 'MCMC', samples: 10000}, model));
```

We'll Do More Examples Together