

Bayesian Zig Zag

Developing probabilistic models
using grid methods and MCMC

Allen Downey
Olin College

ACM Learning Center
February 2019

These slides tinyurl.com/zigzagacm

Bayesian methods

Increasingly important, but...

Bayesian methods

Increasingly important, but...

hard to get started.

Simply conditioning on the known value of the data y , using the basic property of conditional probability known as Bayes' rule, yields the *posterior* density:

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)}, \quad (1.1)$$

where $p(y) = \sum_{\theta} p(\theta)p(y|\theta)$, and the sum is over all possible values of θ (or $p(y) = \int p(\theta)p(y|\theta)d\theta$ in the case of continuous θ). An equivalent form of (1.1) omits the factor $p(y)$, which does not depend on θ and, with fixed y , can thus be considered a constant, yielding the *unnormalized posterior density*, which is the right side of (1.2):

$$p(\theta|y) \propto p(\theta)p(y|\theta). \quad (1.2)$$

The second term in this expression, $p(y|\theta)$, is taken here as a function of θ , not of y . These simple formulas encapsulate the technical core of Bayesian inference: the primary task of any specific application is to develop the model $p(\theta, y)$ and perform the computations to summarize $p(\theta|y)$ in appropriate ways.

Prediction

To make inferences about an unknown observable, often called predictive inferences, we follow a similar logic. Before the data y are considered, the distribution of the unknown but observable y is

$$p(y) = \int p(y, \theta)d\theta = \int p(\theta)p(y|\theta)d\theta. \quad (1.3)$$

This is often called the marginal distribution of y , but a more informative name is the *prior predictive distribution*: prior because it is not conditional on a previous observation of the process, and predictive because it is the distribution for a quantity that is observable.

After the data y have been observed, we can predict an unknown observable, \tilde{y} , from the same process. For example, $y = (y_1, \dots, y_n)$ may be the vector of recorded weights of an object weighed n times on a scale, $\theta = (\mu, \sigma^2)$ may be the unknown true weight of the object and the measurement variance of the scale, and \tilde{y} may be the yet to be recorded weight of the object in a planned new weighing. The distribution of \tilde{y} is called the *posterior predictive distribution*, posterior because it is conditional on the observed y and predictive because it is a prediction for an observable \tilde{y} :

$$\begin{aligned} p(\tilde{y}|y) &= \int p(\tilde{y}, \theta|y)d\theta \\ &= \int p(\tilde{y}|\theta, y)p(\theta|y)d\theta \\ &= \int p(\tilde{y}|\theta)p(\theta|y)d\theta. \end{aligned} \quad (1.4)$$

The second and third lines display the posterior predictive distribution as an average of conditional predictions over the posterior distribution of θ . The last step follows from the

Bayesian Zig Zag

An approach I think is good for

1. Learning.
2. Developing models iteratively.
3. Validating models incrementally.

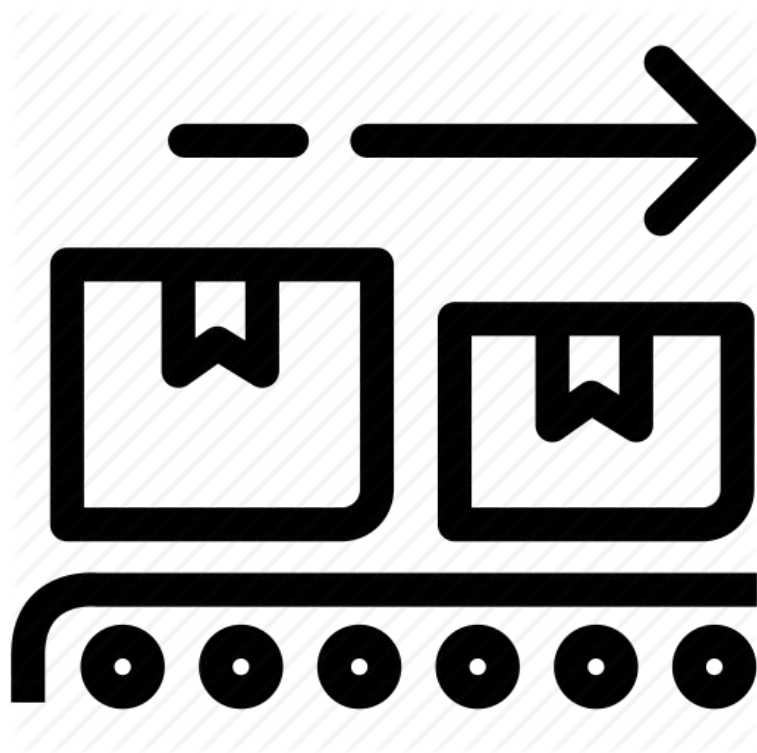
Forward and inverse probability.

Forward probability

You have a model of the system.

You know the parameters.

You can generate data.

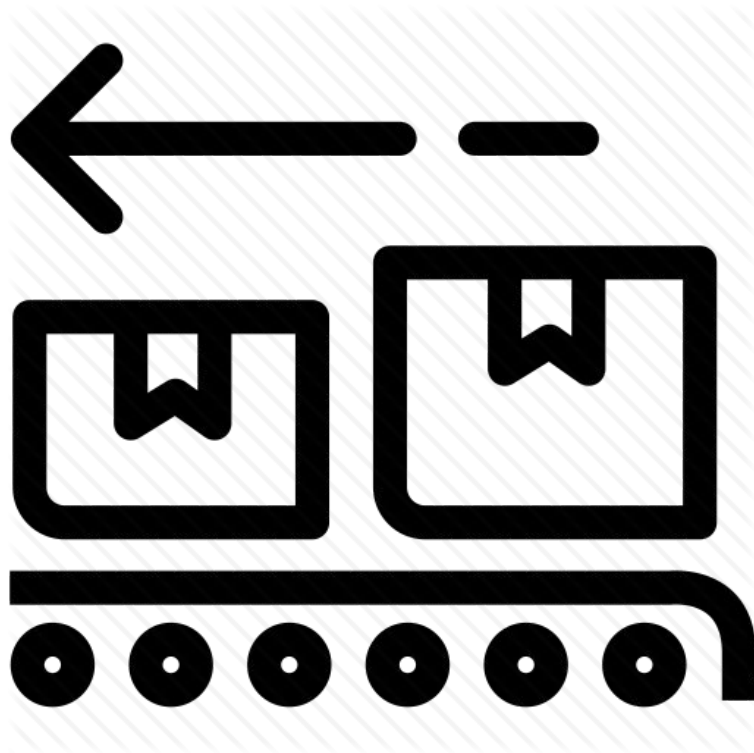


Inverse probability

You have a model of the system.

You have data.

You can estimate the parameters.



Start forward

Simulate the model.

Go backward

Run grid approximations.

Go forward

Generate predictive distributions.

And here is a key...

Go forward

Generate predictive distributions.

Generating predictions looks a lot like a PyMC model.

Go backward

Run the PyMC model.

Validate against the grid approximations.

Go forward

Use PyMC to generate predictions.

Let's look at an example.



BOS 

27-17-5, 59 PTS

ESPN+
2/15
10:00 PM



ANA
21-21-9, 51 PTS



Regular Season Series

BOS leads 1-0

 **Bruins**

Game 2

 **Ducks**

2/15

ESPN+

 **Ducks**

1

Game 1

 **Bruins**

3

12/20

Final

Hockey?

Well, yes.

But also any system well-modeled by a Poisson process.

Poisson process

Events are equally likely to occur at any time.

1. How long until the next event?
2. How many events in a given interval?

Let's get to it

These slides tinyurl.com/zigzagacm

Read the notebook:

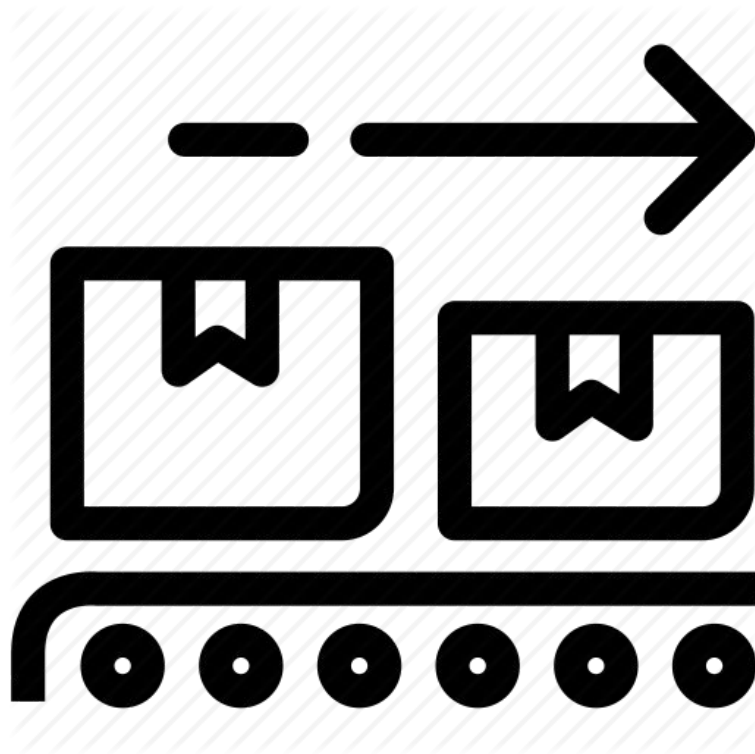
- Static view on [GitHub](#).
- Live on [Binder](#).

I'll use Python code to show:

- Most steps are a few lines of code,
- Based on standard libraries (NumPy, SciPy, PyMC).

Don't panic.

STEP 1: FORWARD



Simulating hockey

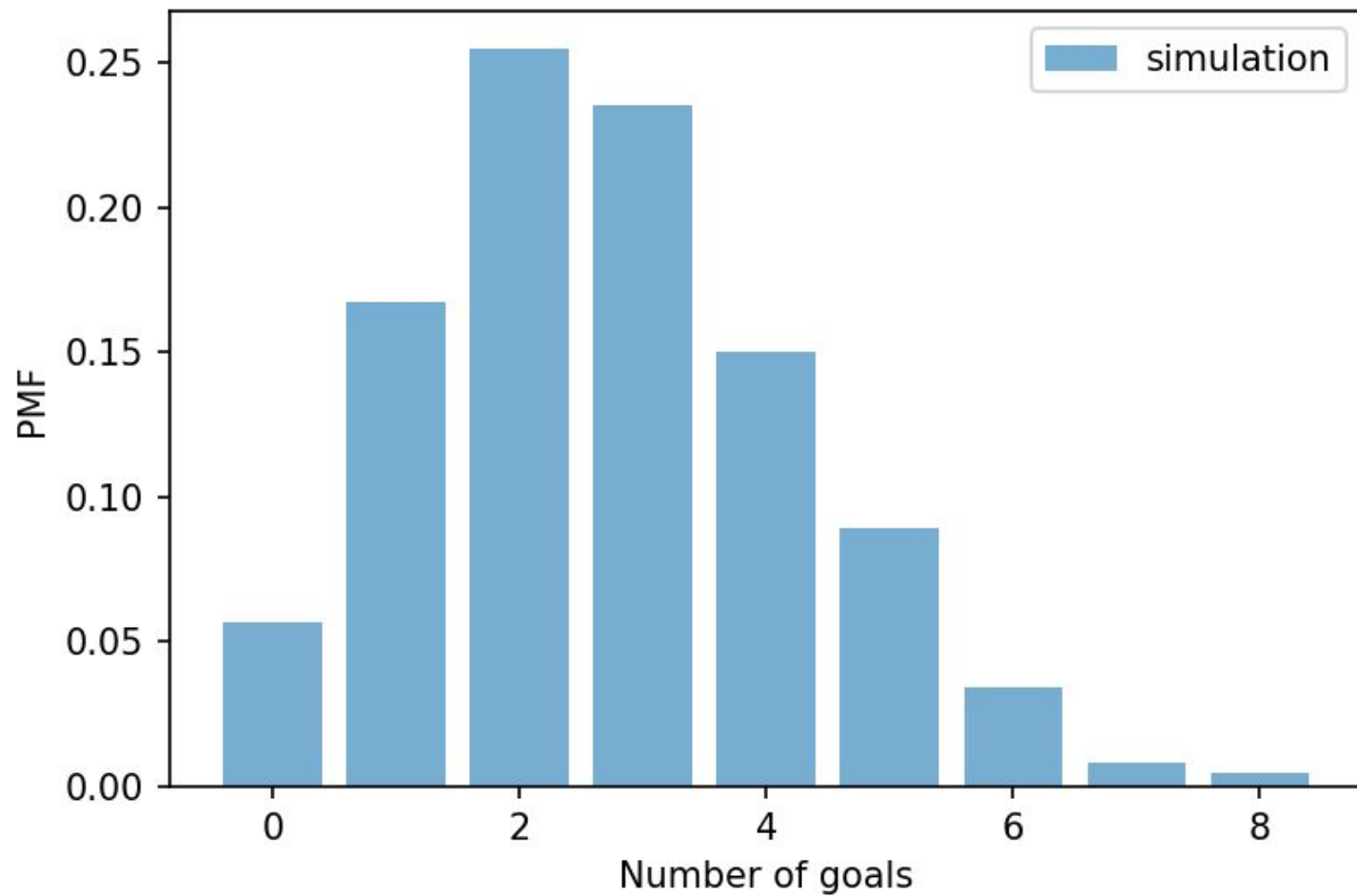
Probability of scoring a goal in any minute is p .

Pretend we know p .

Simulate 60 minutes and add up the goals.

```
def simulate_game(p, n=60):  
    goals = np.random.choice([0, 1], n, p=[1-p, p])  
    return np.sum(goals)
```

Distribution of goals scored



Analytic distributions

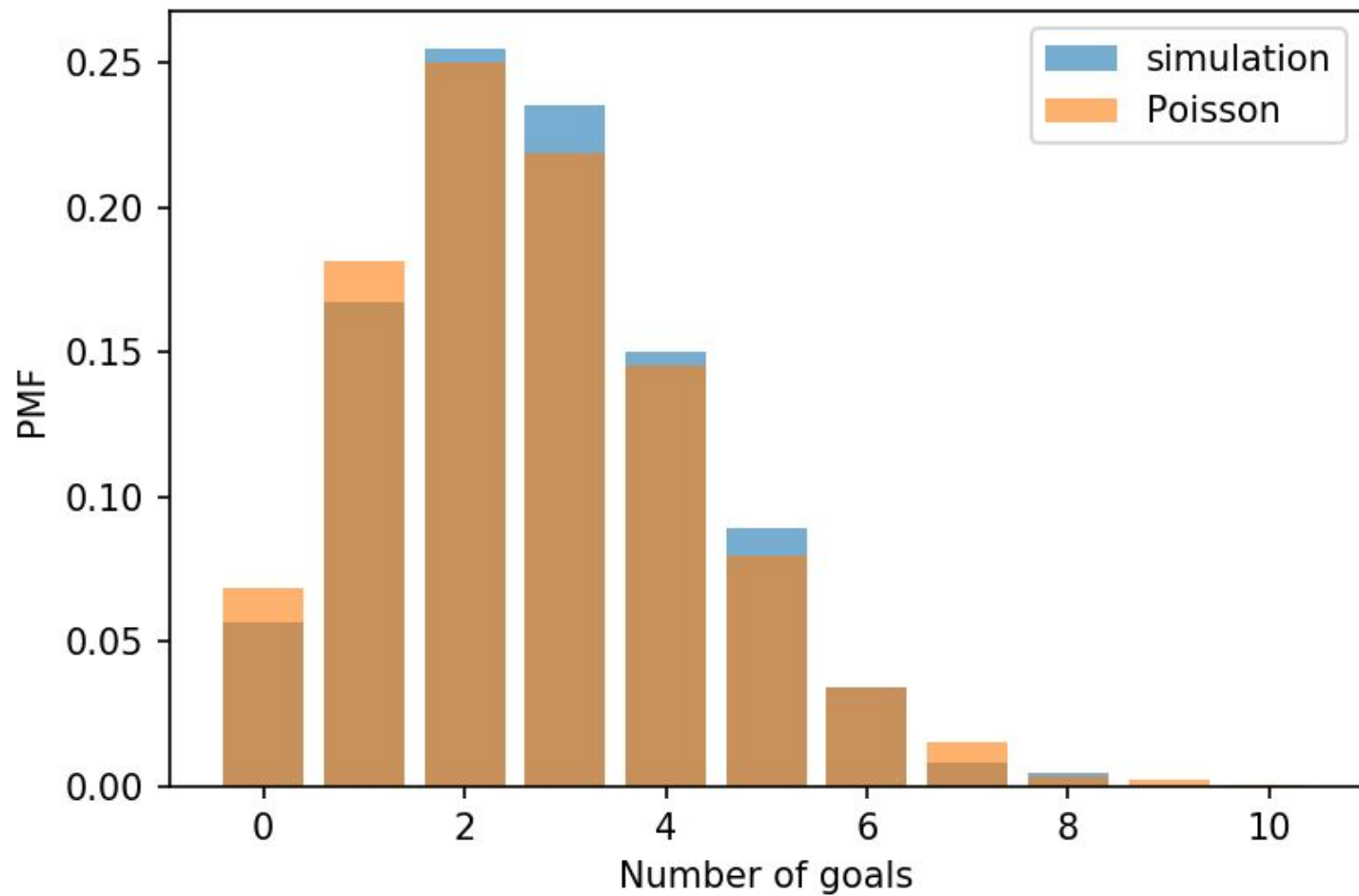
Result of the simulation is binomial.

Well approximated by Poisson.


```
mu = n * p
```

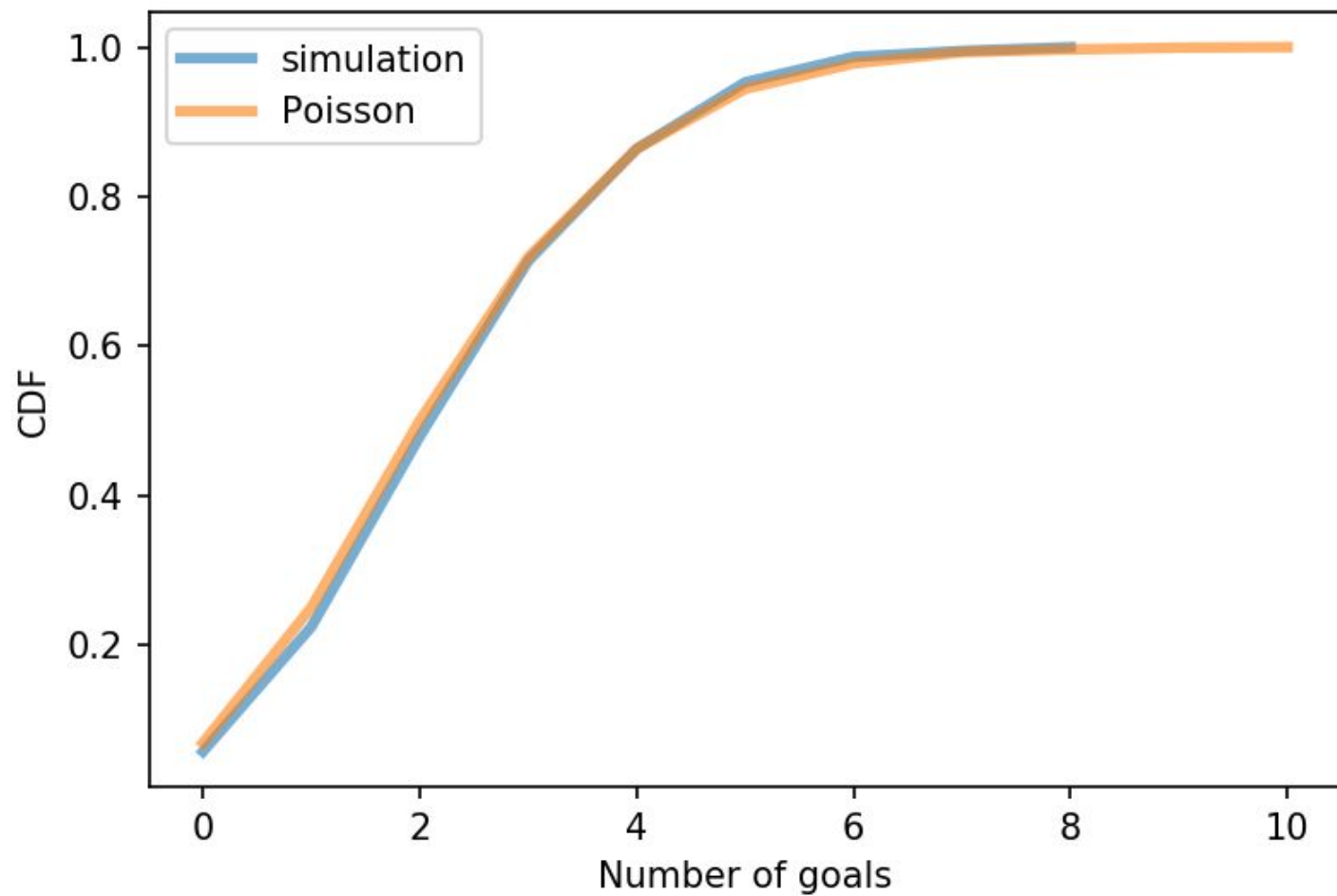
```
sample_poisson = np.random.poisson(mu, 1000)
```

Distribution of goals scored



To compare distributions,
cumulative distribution function (CDF)
is better than
probability mass function (PMF).

Distribution of goals scored



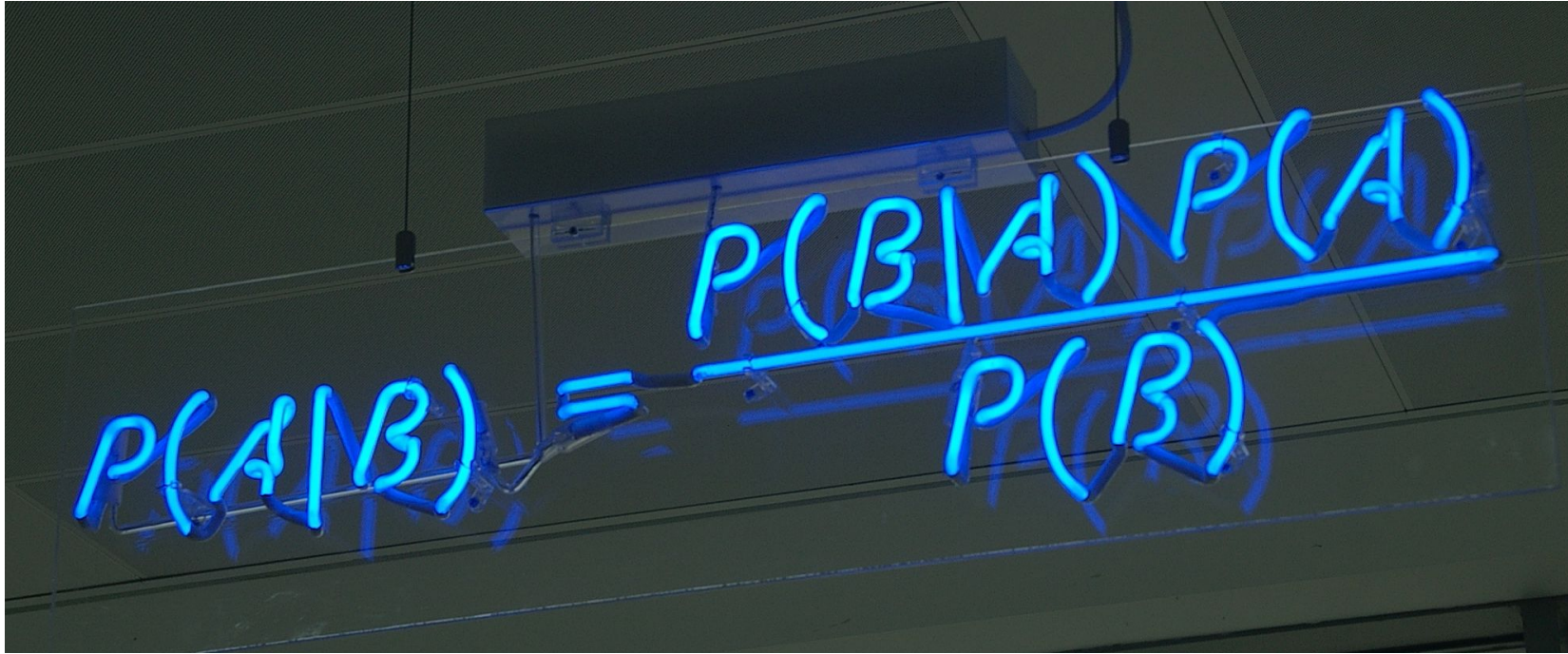
Forward

So far, forward probability.

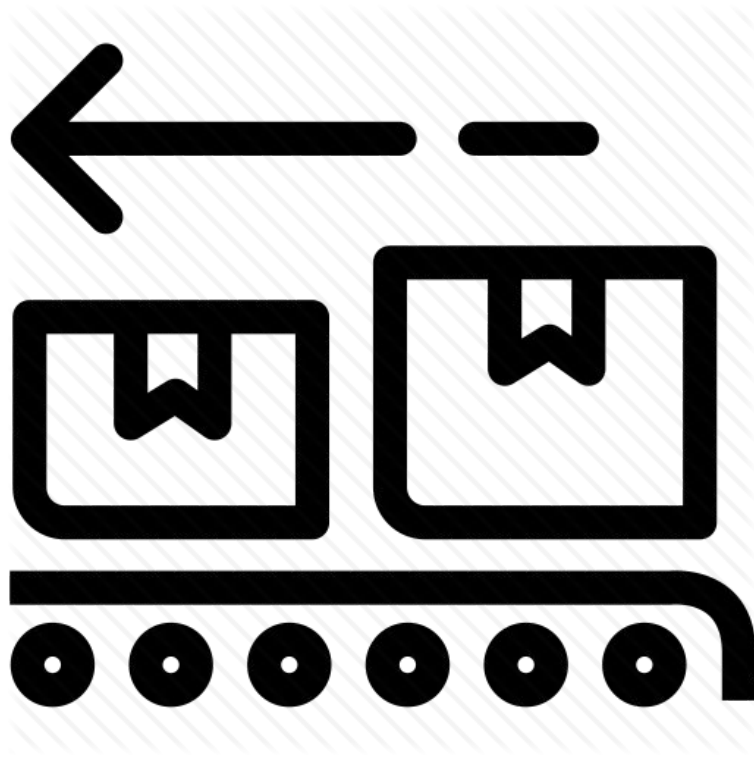
Given μ , we can compute $p(\text{goals} \mid \mu)$.

For inference we want $p(\mu \mid \text{goals})$.

Bayes's theorem tells us how they are related.

A photograph of a whiteboard with the Bayes' theorem formula written in blue neon light. The formula is
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$
 The whiteboard is mounted on a wall, and the neon light is glowing brightly. The background is dark, making the blue neon stand out. The whiteboard has a grid pattern, and the formula is written in a clear, hand-drawn style.
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

STEP 2: INVERSE



Bayesian update

Start with **prior** beliefs, $p(\mu)$, for a range of μ .

Compute the likelihood function, $p(\text{goals} \mid \mu)$

Use Bayes's theorem to get **posterior** beliefs, $p(\mu \mid \text{goals})$.


```
def bayes_update(suite, data, like_func):  
    for hypo in suite:  
        suite[hypo] *= like_func(data, hypo)  
    normalize(suite)
```

`suite`: dictionary with possible values of `mu` and probabilities

`data`: observed number of goals

`like_func`: likelihood function that computes $p(\text{goals} \mid \mu)$

```
from scipy.stats import poisson

def poisson_likelihood(goals, mu):
    """Computes  $p(\text{goals} \mid \mu)$ """
    return poisson.pmf(goals, mu)
```

Gamma prior

Gamma distribution has a reasonable shape for this context.

And we can estimate parameters from past games.

```
alpha = 9
```

```
beta = 3
```

```
hypo_mu = np.linspace(0, 15, num=101)
```

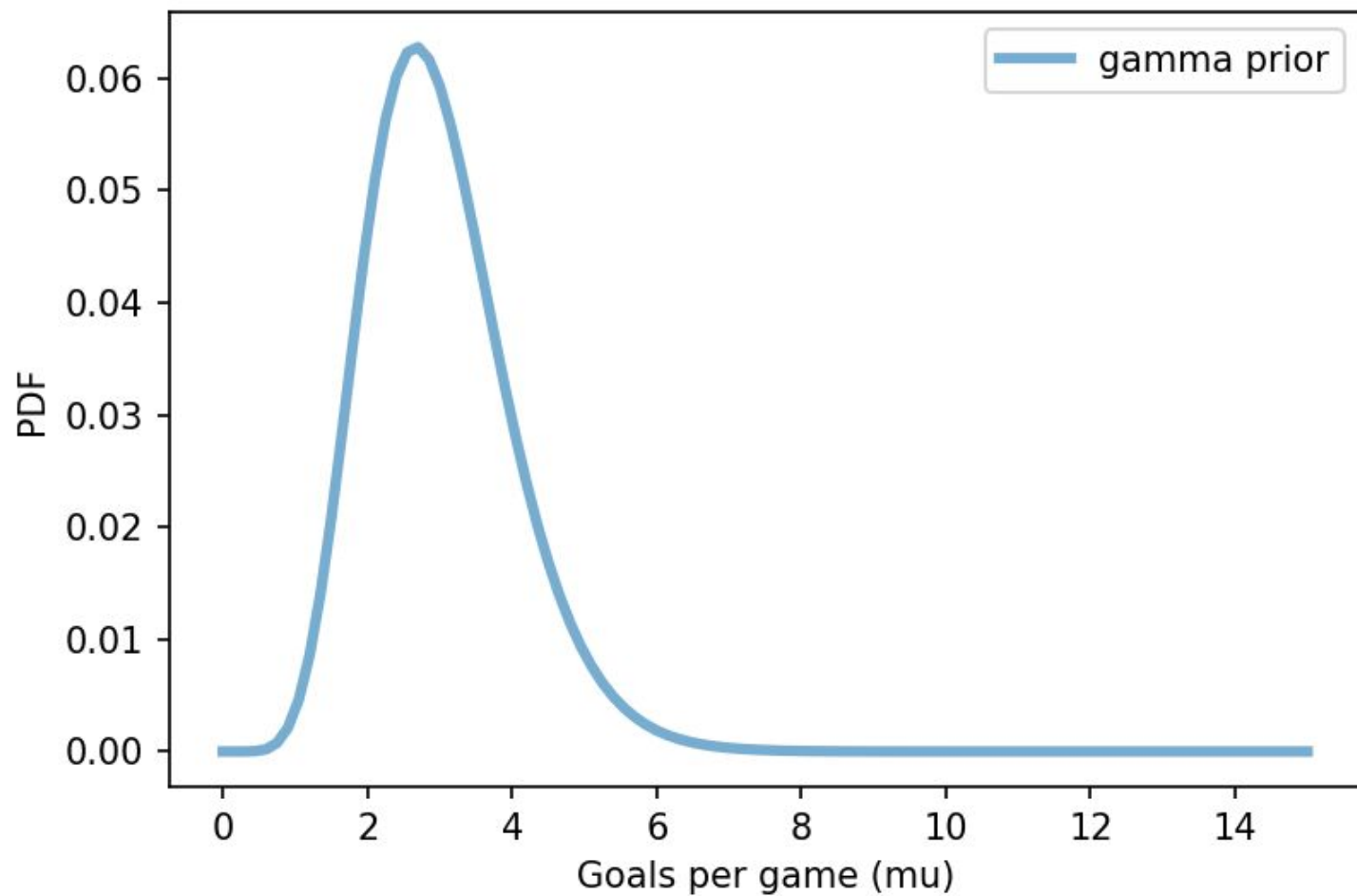
```
gamma_prior = make_gamma_suite(hypo_mu, alpha, beta)
```

Grid approximation

μ is actually continuous.

We're approximating it with a **grid** of discrete values.

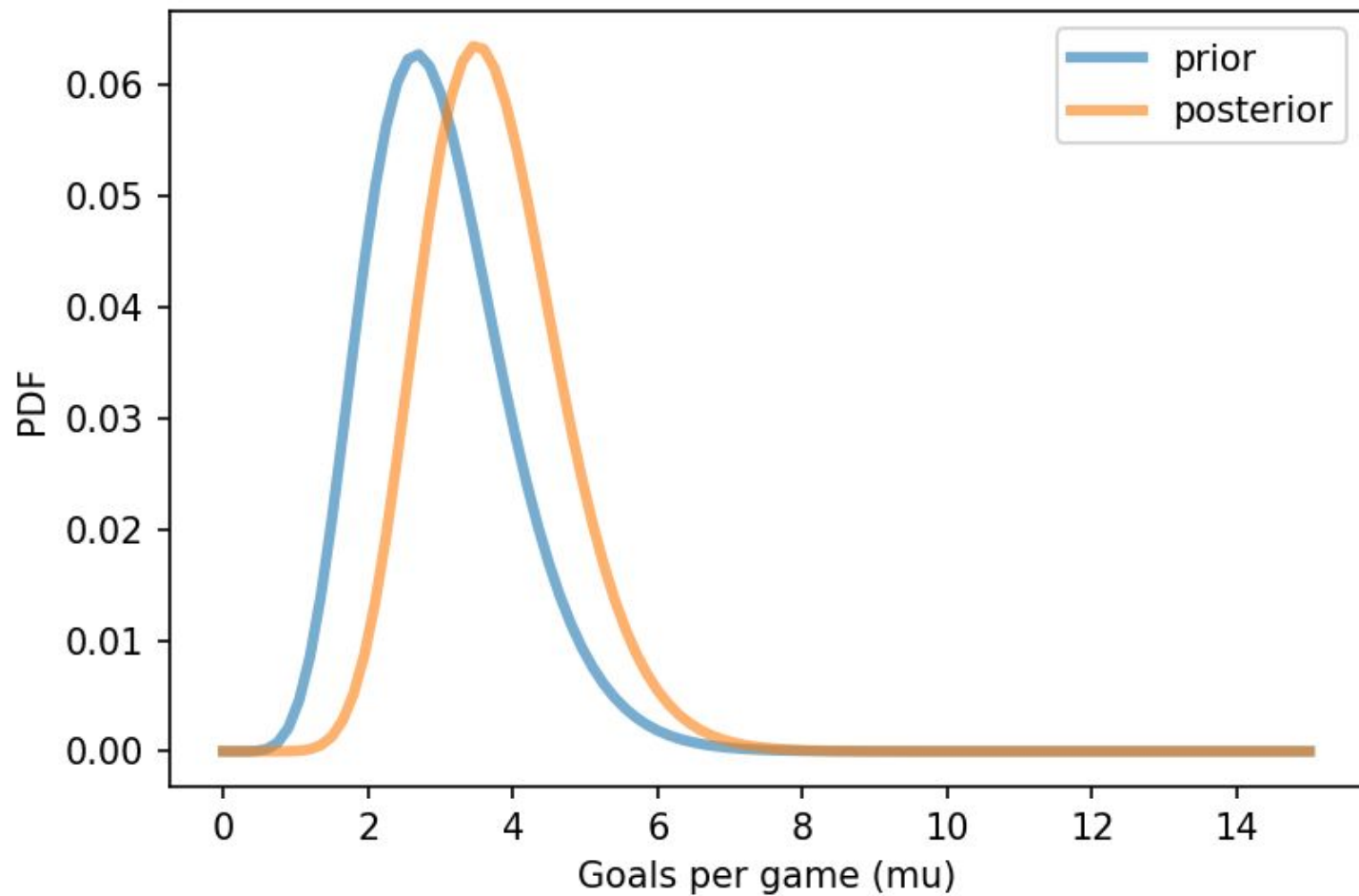
Distribution of goal scoring rate



```
posterior = gamma_prior.copy()
```

```
posterior.bayes_update(data=6, poisson_likelihood)
```

Distribution of goal scoring rate

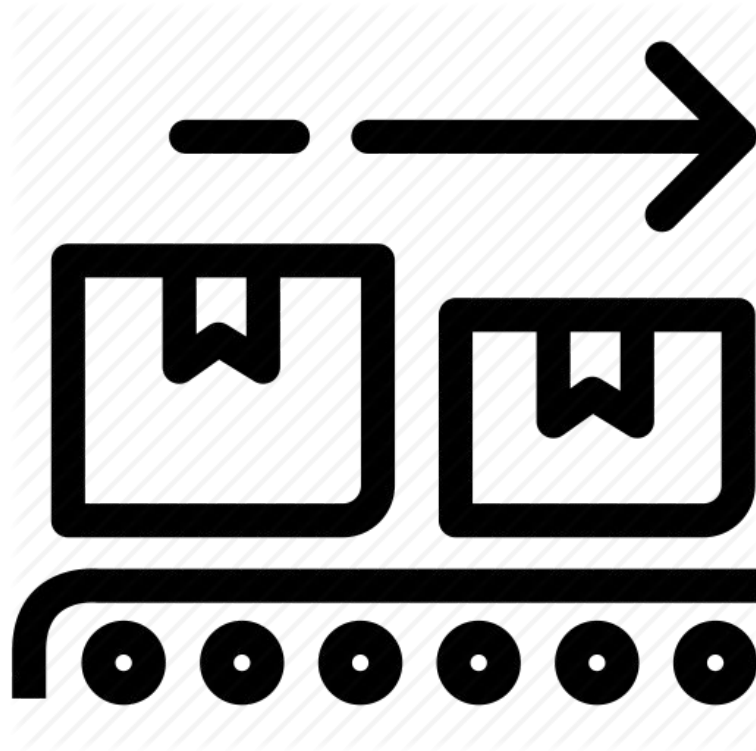


From posterior to predictive

Posterior distribution
represents what we know about μ .

Posterior **predictive** distribution
represents a prediction about the number of goals.

STEP 3: FORWARD



Sampling

To sample the posterior predictive distribution:

1. Draw random μ from the posterior.
2. Draw random $goals$ from $Poisson(\mu)$.
3. Repeat.

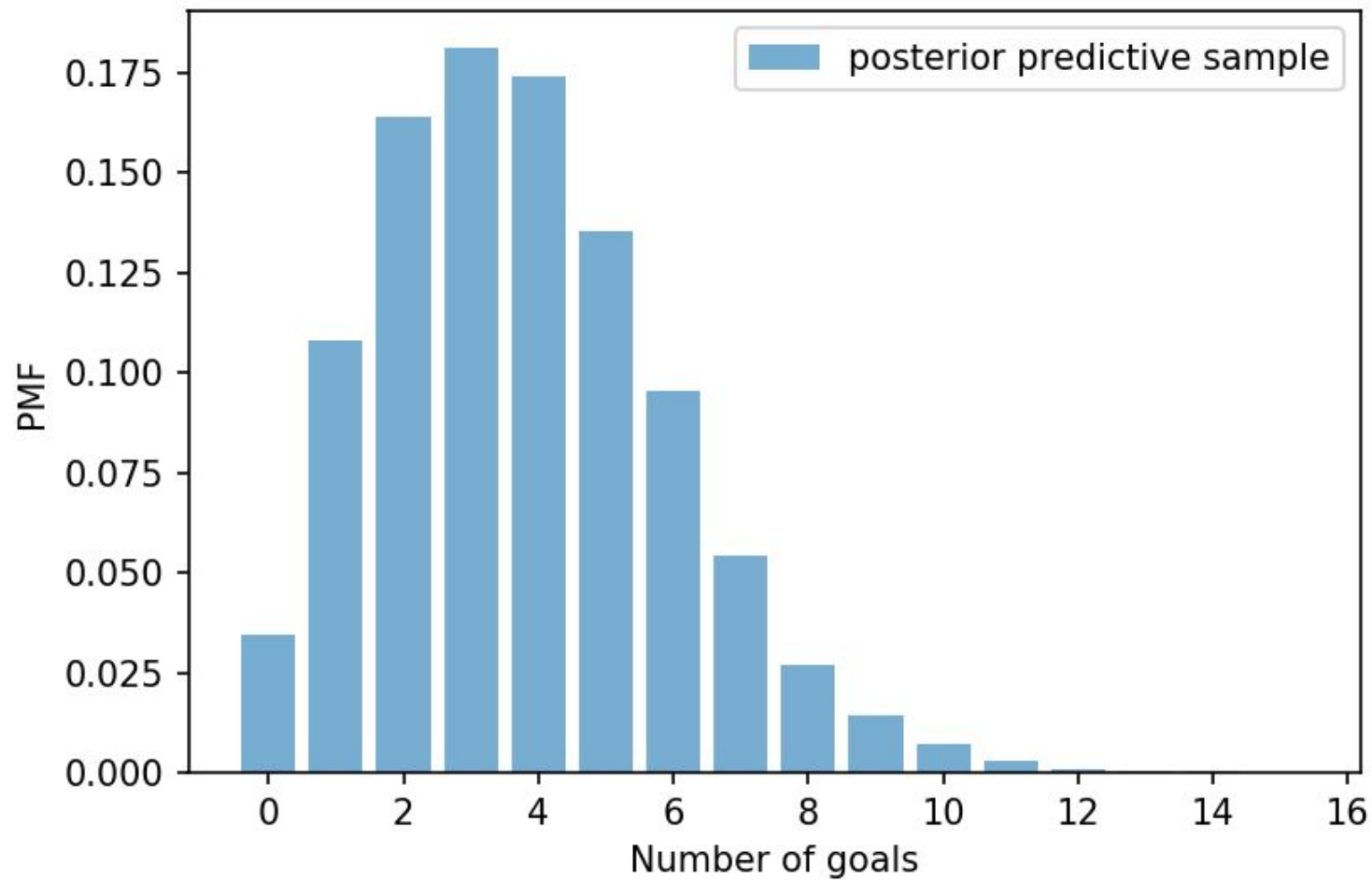
```
def sample_suite(suite, n):  
    mus, p = zip(*suite.items())  
    return np.random.choice(mus, n, replace=True, p=p)
```

`suite`: dictionary with possible values of `mu` and probabilities

```
sample_post = sample_suite(posterior, n)
```

```
sample_post_pred = np.random.poisson(sample_post)
```

Distribution of goals scored



Posterior predictive distribution

Represents two sources of uncertainty:

1. We're unsure about μ .
2. Even if we knew μ , we would be unsure about goals.

Forward PyMC

I'll use PyMC to run the forward model.

Overkill, but it helps:

- Validate: does the model make sense?
- Verify: did we implement the model we intended?


```
model = pm.Model()
```

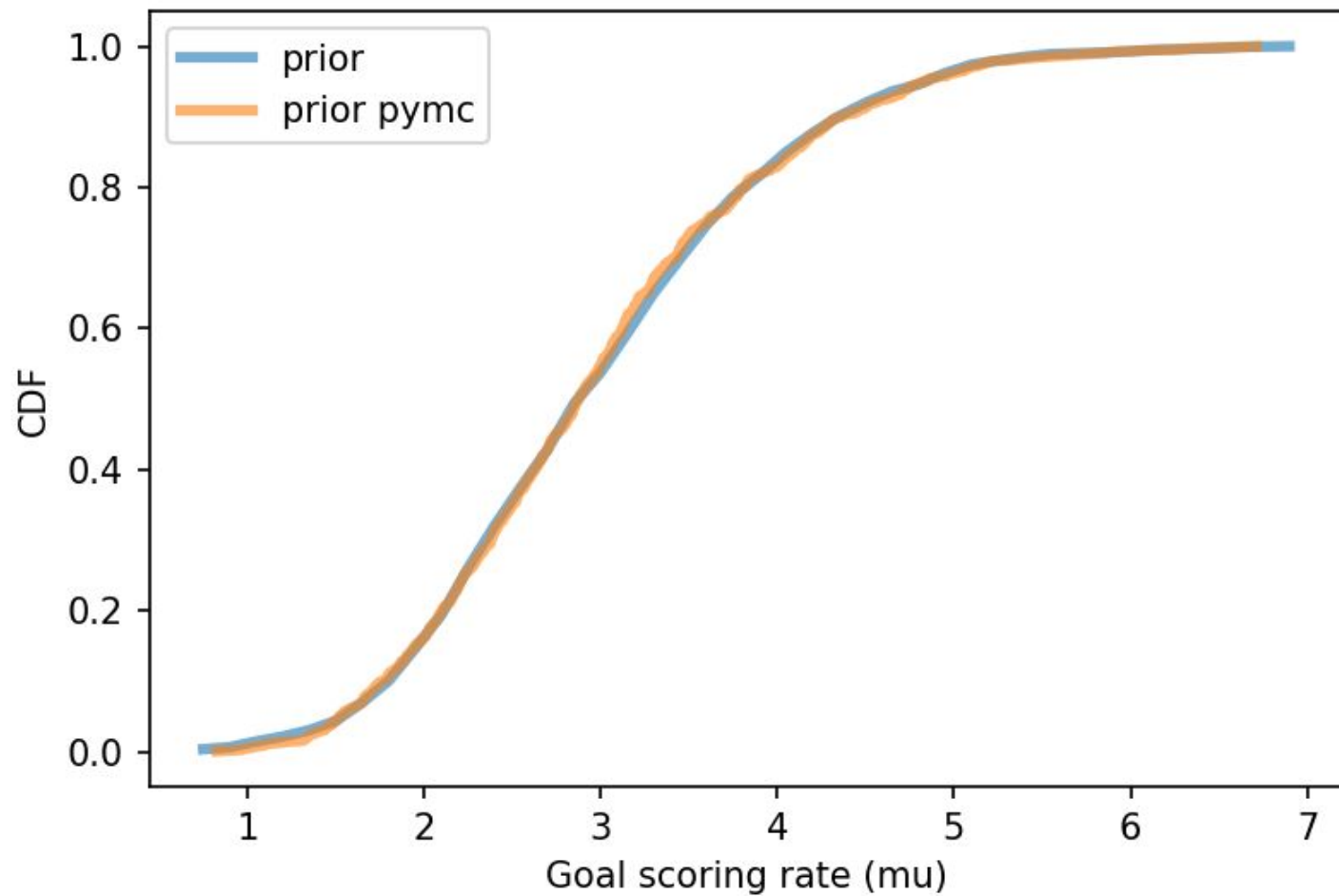
with model:

```
mu = pm.Gamma('mu', alpha, beta)
```

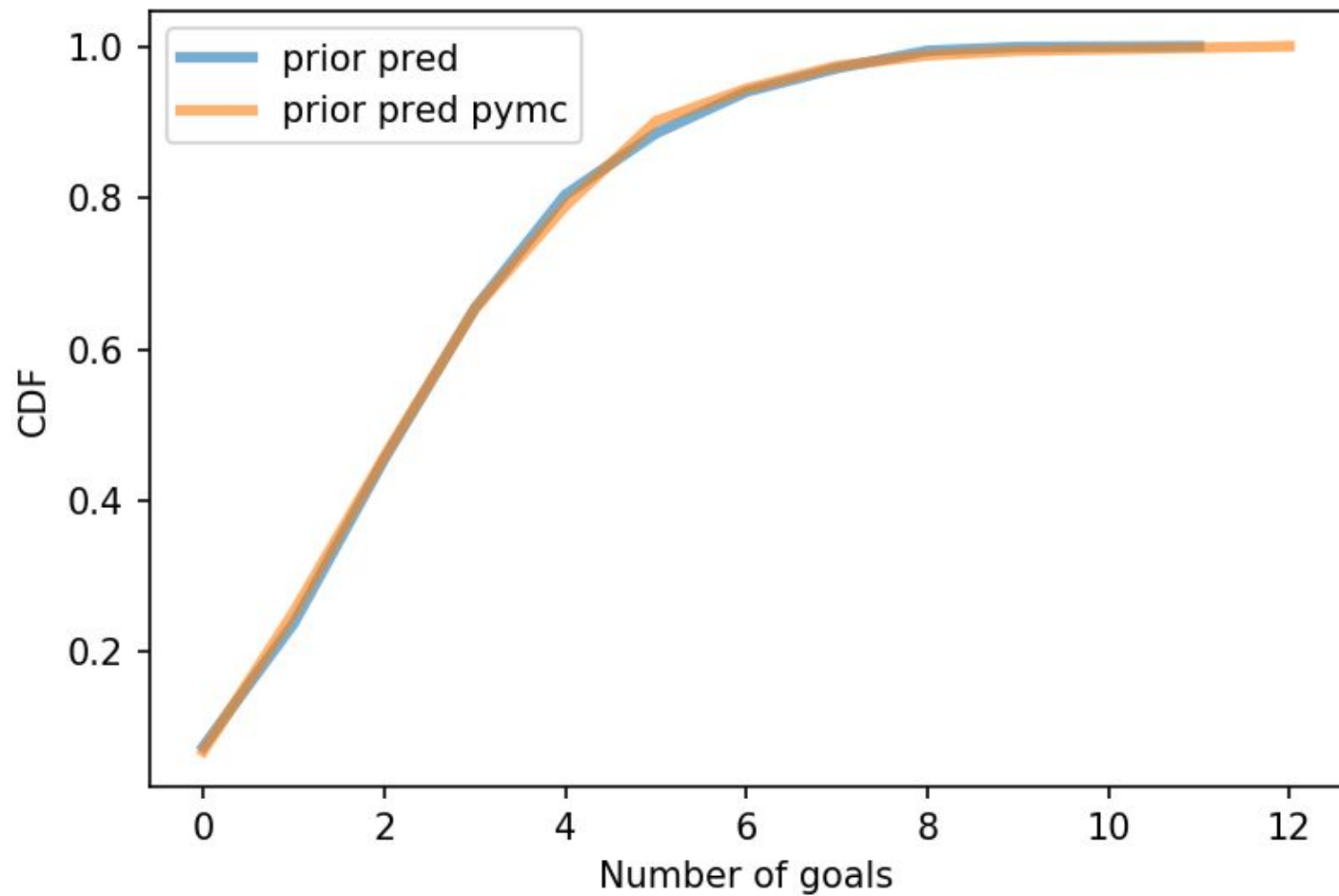
```
goals = pm.Poisson('goals', mu)
```

```
trace = pm.sample_prior_predictive(1000)
```

Distribution of goal scoring rate



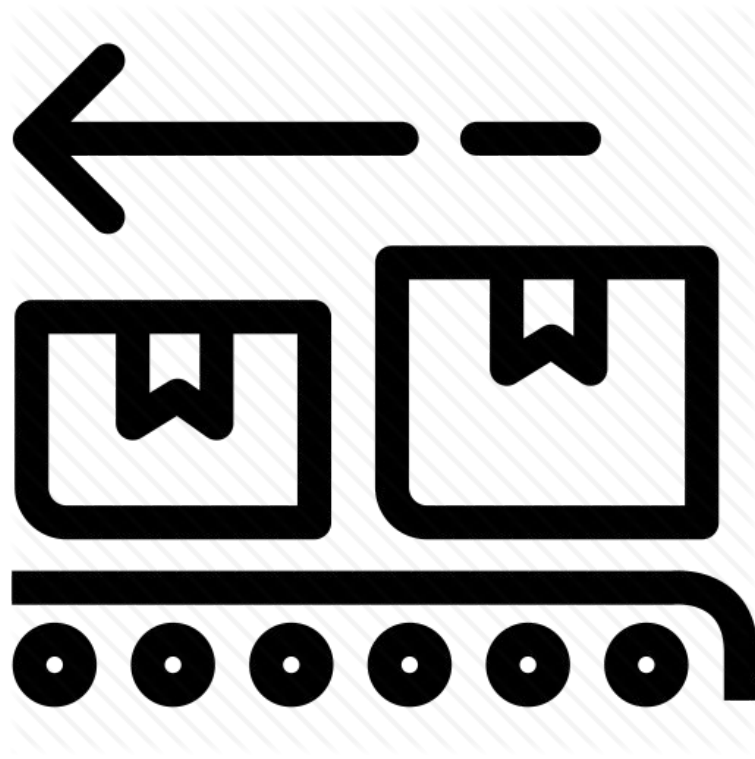
Distribution of goals scored



This confirms that we specified the model right.

And it helps with the next step.

STEP 4: INVERSE



```
model = pm.Model()
```

with model:

```
mu = pm.Gamma('mu', alpha, beta)
```

```
goals = pm.Poisson('goals', mu)
```

```
trace = pm.sample_prior_predictive(1000)
```

```
model = pm.Model()
```

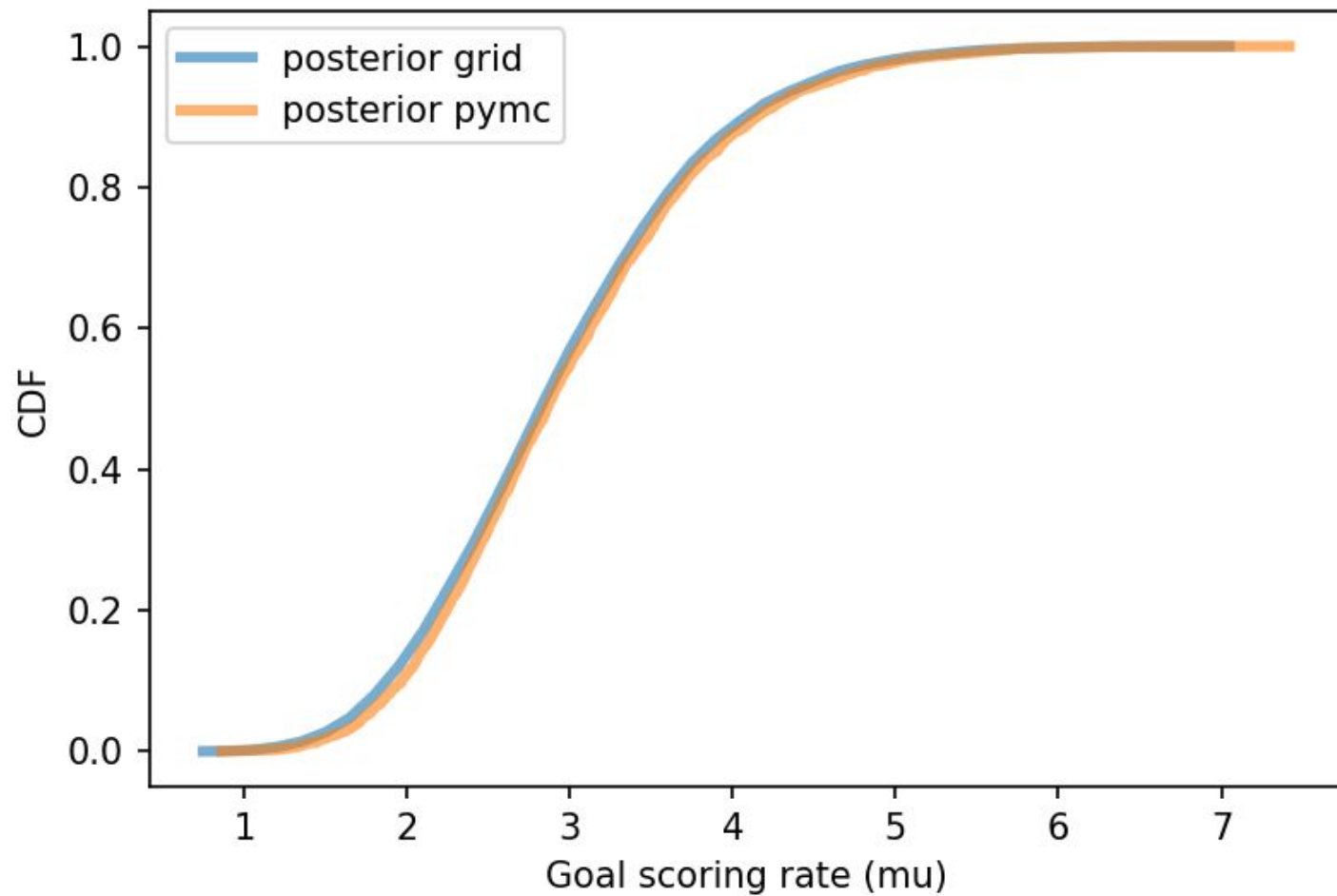
with model:

```
mu = pm.Gamma('mu', alpha, beta)
```

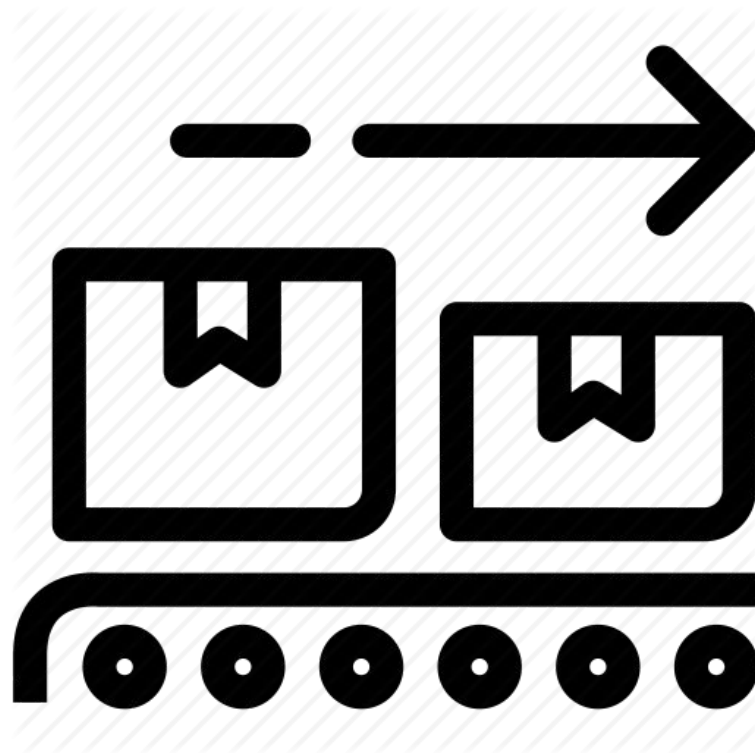
```
goals = pm.Poisson('goals', mu, observed=3)
```

```
trace = pm.sample(1000)
```

Distribution of goal scoring rate

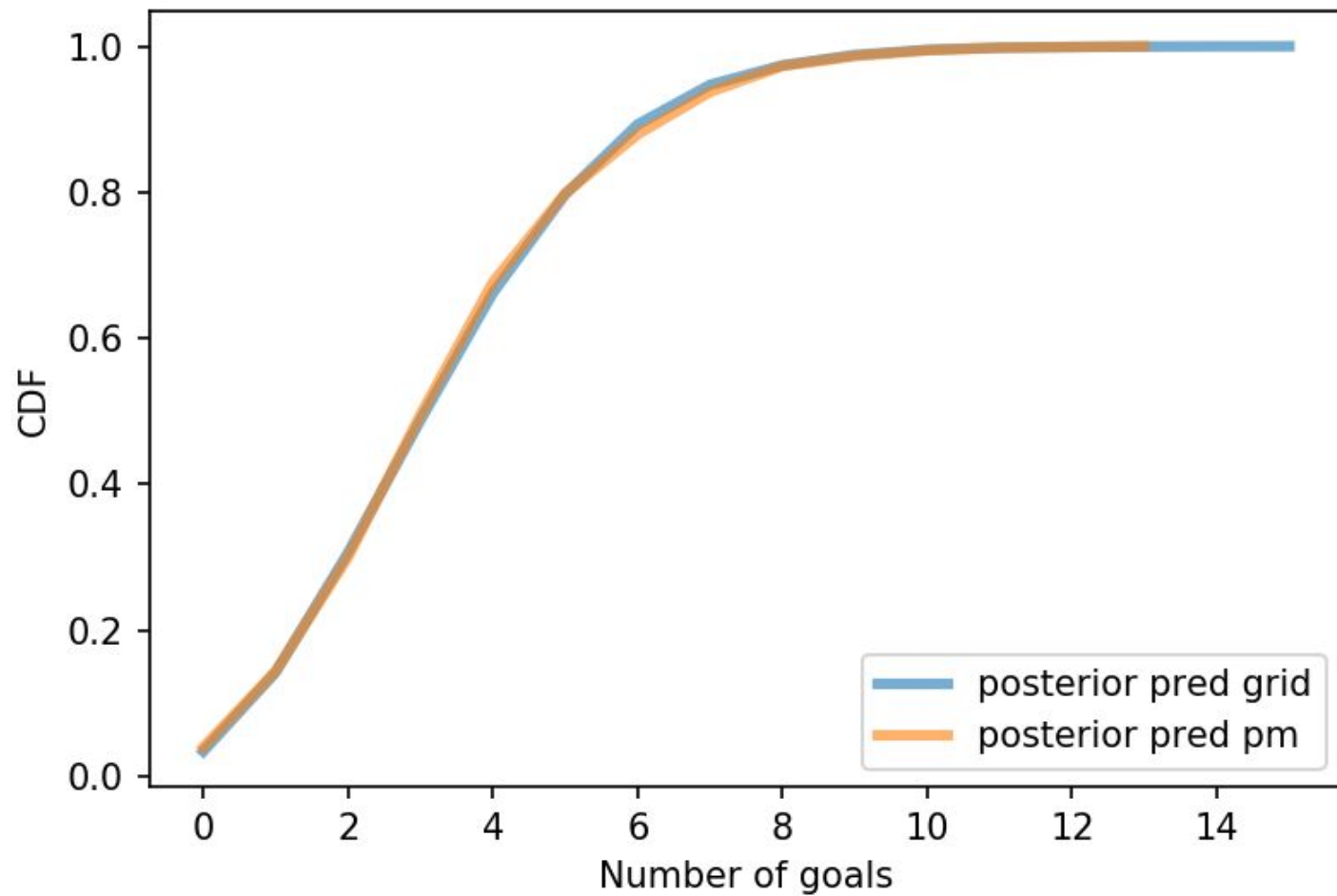


STEP 5: FORWARD



```
post_pred = pm.sample_posterior_predictive(trace, samples=1000)
```

Distribution of goals scored



With a working PyMC model,
we can take on problems too big for
grid algorithms.



BOS 

27-17-5, 59 PTS

ESPN+
2/15
10:00 PM



ANA
21-21-9, 51 PTS



Regular Season Series

BOS leads 1-0

 **Bruins**

Game 2

 **Ducks**

2/15

ESPN+

 **Ducks**

1

Game 1

 **Bruins**

3

12/20

Final

Two teams

Starting with the same prior:

- Update BOS with observed=3.
- Update ANA with observed=1.

```
model = pm.Model()
```

with model:

```
mu_BOS = pm.Gamma('mu_BOS', alpha, beta)
```

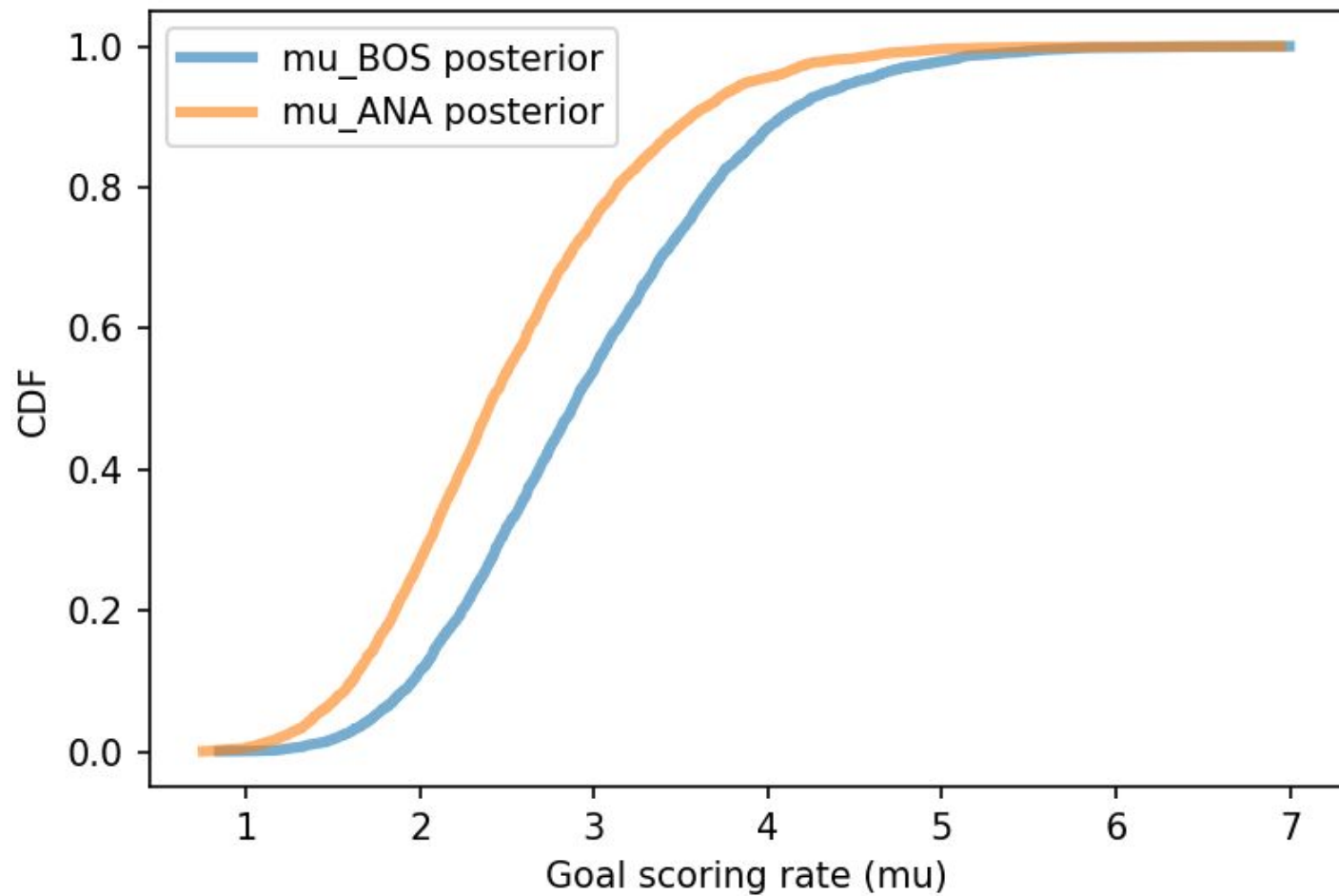
```
mu_ANA = pm.Gamma('mu_ANA', alpha, beta)
```

```
goals_BOS = pm.Poisson('goals_BOS', mu_BOS, observed=3)
```

```
goals_ANA = pm.Poisson('goals_ANA', mu_ANA, observed=1)
```

```
trace = pm.sample(1000)
```

Distribution of goal scoring rate



Probability of superiority

```
mu_BOS = trace['mu_BOS']
```

```
mu_ANA = trace['mu_ANA']
```

```
np.mean(mu_BOS > mu_ANA)
```

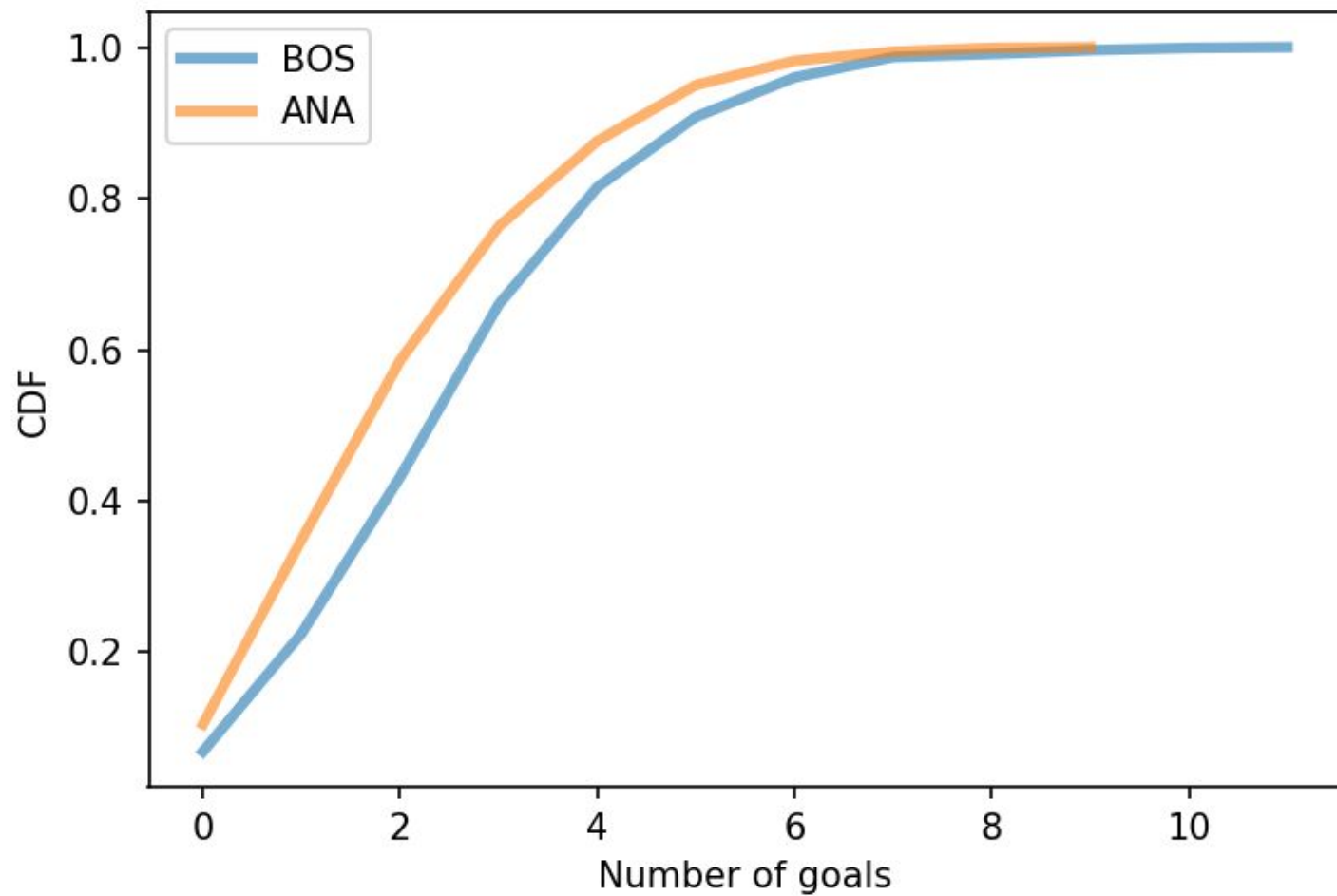
```
0.67175
```

```
post_pred = pm.sample_posterior_predictive(trace, samples=1000)
```

```
goals_BOS = post_pred['goals_BOS']
```

```
goals_ANA = post_pred['goals_ANA']
```

Distribution of goals scored



Probability of winning

```
win = np.mean(goals_BOS > goals_ANA)
```

```
0.488
```

```
lose = np.mean(goals_ANA > goals_BOS)
```

```
0.335
```

```
tie = np.mean(goals_BOS == goals_ANA)
```

```
0.177
```

Overtime!

Time to first goal is exponential with $1/\mu$.

Generate predictive samples.

```
tts_BOS = np.random.exponential(1/mu_BOS)
```

```
tts_ANA = np.random.exponential(1/mu_ANA)
```

```
win_ot = np.mean(tts_BOS < tts_ANA)
```

```
0.55025
```

```
total_win = win + tie * win_ot
```

```
0.58539425
```

Summary



Think Bayes

Chapter 7:

The Boston Bruins problem

Available under a free license
at thinkbayes.com.

And published by O'Reilly Media.

Bayesian Statistics in Python

Think Bayes



O'REILLY®

Allen B. Downey

Please don't use this to gamble

First of all, it's only based on data from one previous game.

Also...

Please don't use this to gamble

Gambling a zero-sum game (or less).

If you make money,
you're just taking it from someone else.

As opposed to creating value.

If you made it this far, you probably have some skills.

Use them for better things than gambling.

<https://opendatascience.com/data-science-for-good-part-1/>

Data Science for Good, Part 1

DATA SCIENCE FOR GOOD TECH UPDATES posted by Diego Arenas, ODSC © January 24, 2018

Introduction

This is the first a three-article series about Data Science for Good. This article explains what what this idea is about and how you can get involved in it. The **second article** we'll introduce people, organizations, and projects that use data science for good. The third and **last article** discusses resources and technological tools that serve that purpose.

And finally...

Thanks

Chris Fonnesbeck for help getting these examples running.

Colin Carroll for adding `sample_prior_predictive`.

Eric Ma for moderating today,
and for contributions to PyMC.

These slides: tinyurl.com/zigzagacm

website

github

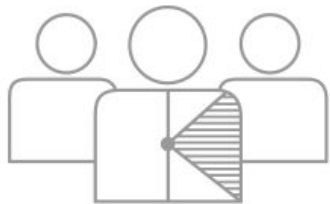
downey@allendowney.com

twitter

email

DataKind
USING DATA IN THE SERVICE OF HUMANITY

Choose Your Own Adventure



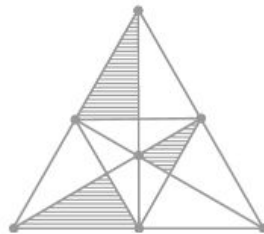
Community Events

Networking and quick consultation to help organizations begin their data science journey.



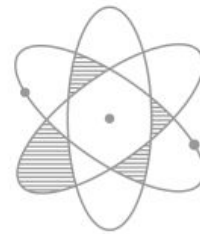
DataDives

Weekend-long, marathon-style events that help organizations do initial data analysis, exploration, and prototyping.



DataCorps

Long-term engagements that help organizations use data science to transform their work and their sector.



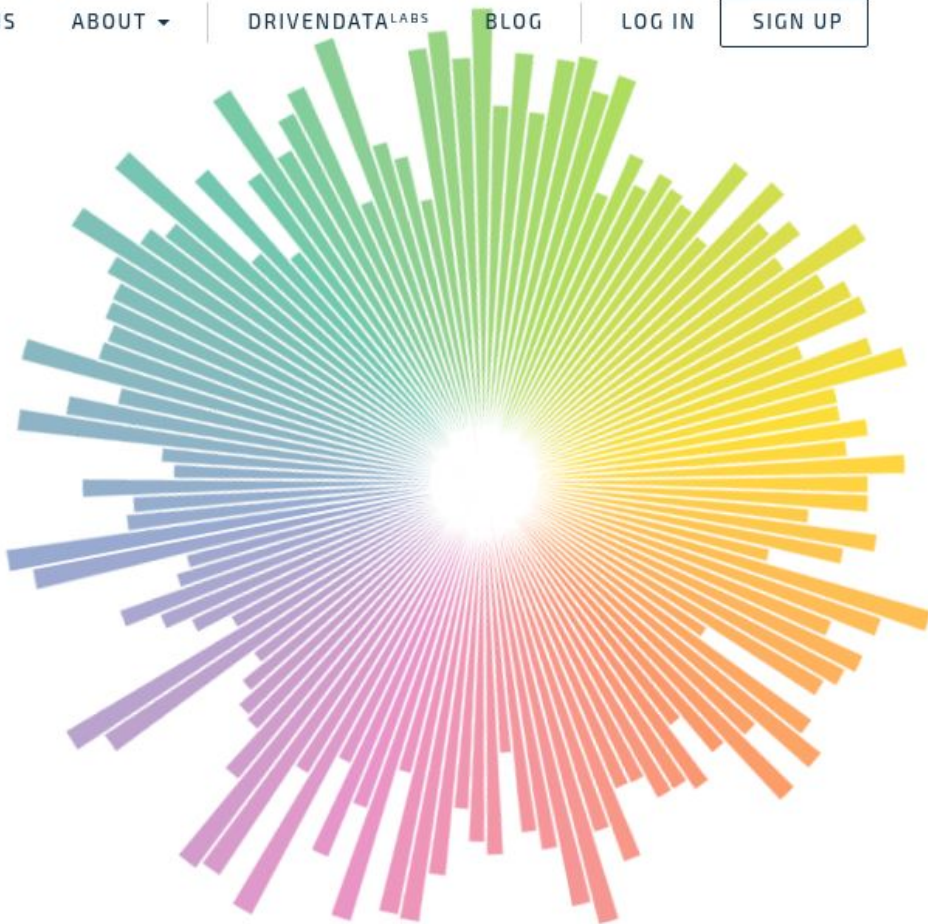
DataKind Labs

Long-term projects that convene multiple stakeholders to develop cutting-edge, cross-sector solutions.

Data science competitions to save the world

[I want to join a competition →](#)

[I want to run a competition →](#)



Search

Categories

DATA SCIENCE NEWS (61)

KAGGLE NEWS (138)

KERNELS (42)

OPEN DATASETS (10)

TUTORIALS (50)

UNCATEGORIZED (3)

WINNERS' INTERVIEWS (220)

Want to subscribe?

Email Address*

First Name

Last Name

Introducing Data Science for Good Events on Kaggle

Megan Risdal | 11.16.2017



Today, we're excited to announce [Kaggle's Data Science for Good program](#)! We're launching the Data Science for Good program to enable the Kaggle community to come together and make significant contributions to tough social good problems with datasets that don't necessarily fit the tight constraints of our traditional supervised machine learning competitions.

What does a Data Science for Good Event Look Like?

[Data Science for Good events](#) will unite the energy and talent of a diverse community to drive positive impact on data problems posed by non-profit hosts. [Kaggle's Datasets platform](#) will provide a democratized workspace for data scientists to analyze the data and publish their work. The open and collaborative environment will encourage data scientists to build on each other's work and to push each problem to the limit of what is possible.

Want to improve your community? [Apply for the Code for America Community Fellowship by April 30th.](#)



Search



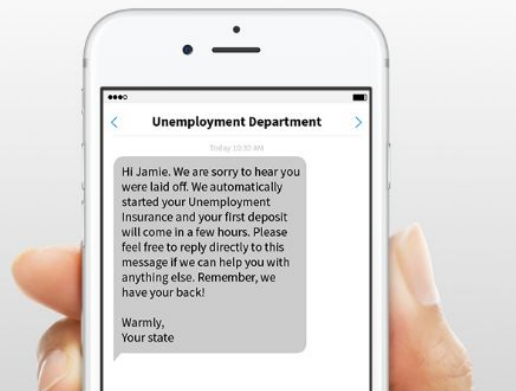
Donate

[Who we are](#) [What we do](#) [How we do it](#) [Join us](#) [Blog](#)

[Work in government](#) [2018 Summit](#)

What if all government services were this good?

The two biggest levers for improving people's lives at scale are technology and government. We put them together.



We're on a mission to make government work in the digital age. Join us.

Work in government



Use your tech and design talents to transform government and impact lives.

[Find open jobs](#)

Help your community



Find your Brigade chapter and work on local projects that matter.

[Volunteer now](#)



POLITICS 05/16/2018 02:43 pm ET

San Francisco To Adopt Software App To Help Automatically Clear Old Marijuana Convictions

“When the government uses 20th-century tools to tackle 21st-century problems, it’s the public that pays the price.”

CODE *for*
AMERICA

UW Data Science for Social Good

The Data Science for Social Good summer program brings together students, stakeholders, data and domain researchers to work on focused, collaborative projects for societal benefit.



Student Fellows – *applications for 2018 closed*

Sixteen DSSG Student Fellows will be selected to work on data-intensive projects that have concrete relevance and social impact. Students are expected to work closely and collaboratively with team members onsite for the duration of the 10-week (June 11- Aug 17) program.

Project Proposals – *submissions now closed*

We invite proposals for 10-week data-intensive research projects to be undertaken in the summer of 2018. We welcome proposals submitted by academic researchers, public agencies, non-profit entities, and industry.

[LEARN MORE ABOUT SUBMITTING A](#)

Project Summaries

Data Science for Social Good projects have an applied social good dimension and broadly address questions related to social science, human services, public policy, criminal justice, environmental impacts, and urban Informatics. Click below to check out previous projects!

[Summer 2017](#)

Data Science For Social Good

Summer Fellowship



THE UNIVERSITY OF
CHICAGO

We're training data scientists to tackle problems that really matter.

APPLY TO BE A
FELLOW, MENTOR,
OR PROJECT
MANAGER

Deadline: Jan 31

SUBMIT A PROJECT
PROPOSAL

Deadline: Extended to

Feb 15

SIGN UP FOR OUR
MAILING LIST

[Join our mailing list](#) to get updates and/or to attend our events this summer.

In addition to the summer fellows, we also hire for our [year-round team](#) at the University of Chicago.

[Get in touch with us](#) if you want to work with us as a Post-doc, Research Assistant, Data Scientist, or Project Partner outside the summer program.

The **Data Science for Social Good Fellowship** is a University of Chicago summer program to train aspiring data scientists to work on data mining, machine learning, big data, and data science projects with social impact. Working closely with governments and nonprofits, fellows take on **real-world problems** in education, health, energy, public safety, transportation, economic development, international development, and more.



This repository

Search

Pull requests

Issues

Marketplace

Explore



dssg / hitchhikers-guide

Watch

116

Star

187

Fork

86

Code

Issues 14

Pull requests 0

Projects 0

Wiki

Insights

The Hitchhiker's Guide to Data Science for Social Good

tutorial-exercises

data-science

517 commits

4 branches

0 releases

21 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



nanounanue Updated creation scripts

Latest commit 5485d25 9 days ago



curriculum

Updated creation scripts

9 days ago



dssg-manual

Update README.md

a month ago



tech-tutorials/model_eval

add bias exercise csv

6 months ago



.gitignore

added simple ETL example

2 years ago



README.md

Update README.md

7 months ago



README.md

Welcome to the Hitchhiker's Guide to Data Science for Social Good.

Our number one priority at [DSSG](#) is to **train fellows to do data science for social good work**. To this end, we've put together this curriculum, which includes many things you'd find in a data science course or bootcamp, but includes an emphasis on social science, ethics, privacy, and social issues.



- ABOUT
- PROJECTS
- LABS
- NEWS
- CHALLENGES
- PRIVACY
- PARTNERSHIPS
- RESOURCES
- CONTACT
- HOME



BIG DATA AND THE SDGs

How data analytics can support monitoring and progress towards the Sustainable Development Goals.

[Read More /](#)



How data analytics can support monitoring and progress towards the Sustainable Development Goals

- NO POVERTY**
Spending patterns on mobile phone services can provide proxy indicators of income levels.
- ZERO HUNGER**
Crowdsourcing or tracking of food prices listed online can help monitor food security in near real-time.
- GOOD HEALTH & WELL-BEING**
Mapping the movement of mobile phone users can help predict the spread of infectious disease.
- QUALITY EDUCATION**
Citizen reporting can reveal reasons for student drop-out rates.
- GENDER EQUALITY**
Analysis of financial transactions can reveal the spending patterns and different impacts of economic shocks on men and women.



NEWS

Indonesian Government Develops a Monitoring Dashboard for the SDGs

Dwayne Carruthers, *Communications Specialist, Pulse Lab Jakarta* Apr 4, 2018

Setting national goals and implementing a set of strategies to achieve them have been central to how modern governments operate. From maintaining economic stability to promoting social welfare, these...

[Read More](#)



TWITTER

Global Pulse Retweeted

+SocialGood @plus_socialgood

What can we learn about gender financial inclusion from micro-finance data? @UNGlobalPulse takes a look: trib.al/C00W7c6

SUBSCRIBE TO OUR NEWSLETTER

email address

GO



Data Science for Good, Part 1

[DATA SCIENCE FOR GOOD](#) [TECH UPDATES](#) posted by [Diego Arenas, ODSC](#) @January 24, 2018

Introduction

This is the first a three-article series about Data Science for Good. This article explains what what this idea is about and how you can get involved in it. The [second article](#) we'll introduce people, organizations, and projects that use data science for good. The third and last article discusses resources and technological tools that serve that purpose.