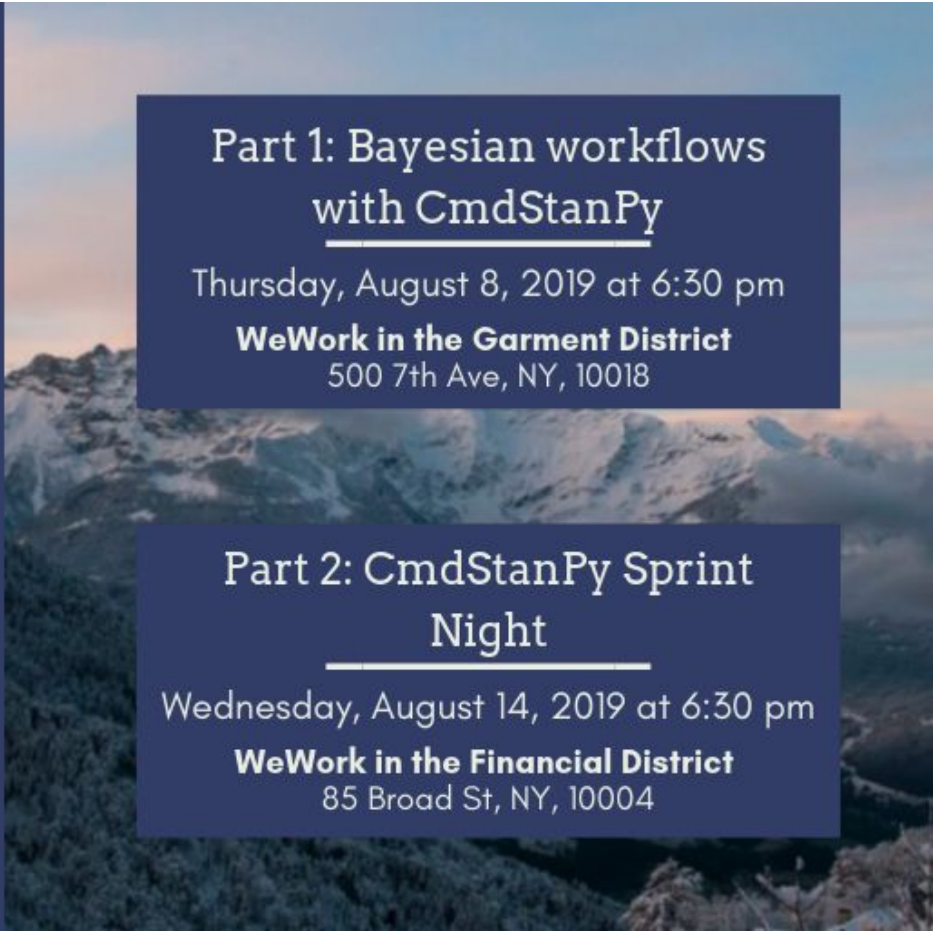
# NYC PyLadies & NYC WiMLDS

along with **Mitzi Morris**
from the Stan development team
present a very special two-part event series!

**Thanks to our host and sponsor:**



## Part 1: Bayesian workflows with CmdStanPy

Thursday, August 8, 2019 at 6:30 pm

**WeWork in the Garment District**
500 7th Ave, NY, 10018

## Part 2: CmdStanPy Sprint Night

Wednesday, August 14, 2019 at 6:30 pm

**WeWork in the Financial District**
85 Broad St, NY, 10004

Follow us on Twitter
@NYCPyLadies

## What is PyLadies?

PyLadies is an international mentorship group with a focus on helping more women become active participants and leaders in the Python open-source community.

Our mission is to promote, educate and advance a diverse Python community through outreach, education, conferences, events and social gatherings.

NYC WiMLDS Meetup
@WiMLDS_NYC

## WiMLDS Mission

To **support and promote women and gender minorities** in machine learning & data science

Membership **inclusive to any person / gender who supports our cause**

WiMLDS is a **501(c)(3) organization**

# Code of Conduct

WiMLDS is dedicated to providing a harassment-free experience for everyone. We do not tolerate harassment of participants in any form.

This code of conduct applies to all WiMLDS spaces, including meetups, Twitter, Slack, mailing lists, both online and offline. Anyone who violates this code of conduct may be sanctioned or expelled from these spaces at the discretion of the Founding Members.

Some WiMLDS spaces may have additional rules in place, which will be made clearly available to participants. Participants are responsible for knowing and abiding by these rules.

**For more information:**

https://github.com/WiMLDS/starter-kit/wiki/Code-of-conduct

# Speaking/Questions

- **Let women speak first.** Given the gender imbalance in tech, this meetup community was created to give women a space and a voice. That includes structuring it in a way that women can do the majority of the speaking, both as speakers and attendees.

- **Q&A time is for asking questions.** If you have a comment and not a question, post it on Twitter.

- **Ask only 1 question at a time** and give other attendees a chance to ask questions. Here is a wonderful piece of advice from Write/Speak/Code:

***"Two people speak once before you speak twice."***

# Get started with CmdStanPy:

- Colab: bit.ly/pyladiesstan3

  Colab will be in “Playground mode” - i.e, you can run the notebook but not make any edits. If you would like to make edits (i.e., “personalize” it), make a copy onto your drive!

- Github: bit.ly/CmdStanPyLadies

  Follow the Readme instructions to get set up!

# Intro to Bayesian Workflows using CmdStanPy

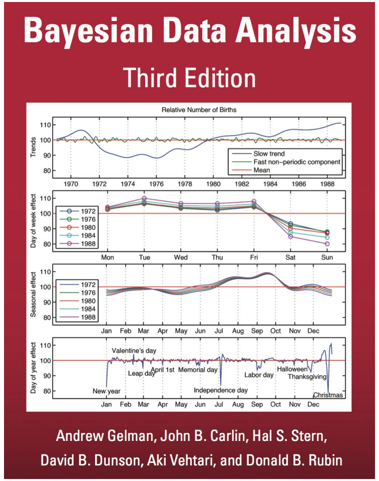
Mitzi Morris

# Talk Outline

- Audience survey
- A few words about Bayesian Data Analysis
- A few words about Stan and CmdStanPy
- Let’s do some Data Analysis!

CHAMPIONS
19

# Bayesian Data Analysis



- *"Statistics is applied statistics and* ***Bayesian data analysis is statistics using conditional probability****"* - Andrew Gelman

- *"By Bayesian data analysis, we mean practical methods for making inferences from data using probability models for quantities we observe and about which we wish to learn.*

  *The essential characteristic of Bayesian methods is their* ***explicit use of probability*** *for* ***quantifying uncertainty in inferences*** *based on statistical analysis."*

  **- Gelman et al., Bayesian Data Analysis, 3rd edition, 2013**

# We wish to learn: WHO’S GONNA WIN???

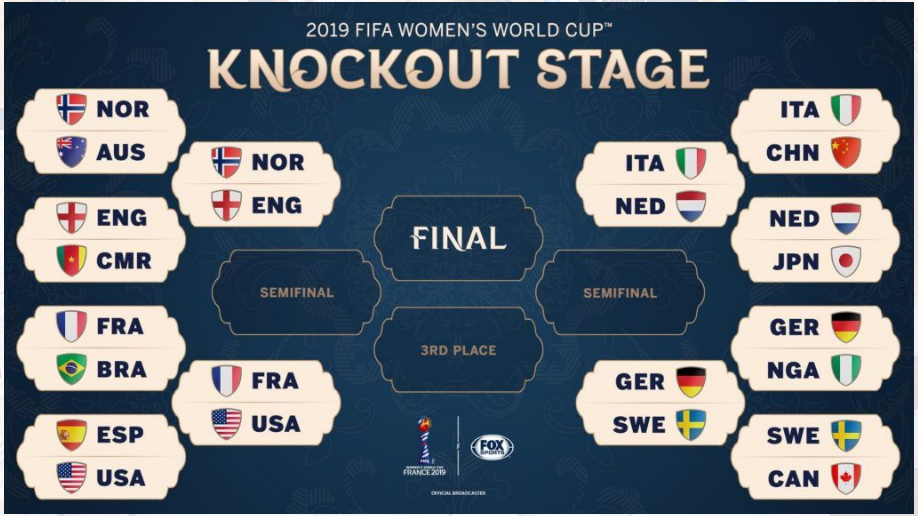
## THE DATA

- **Soccer Power Index (SPI) before the tournament**
  Estimate of team rank going into the World Cup [source]

- **Final scores from all the matches through the quarter finals**

## THE INFERENCE

- **Scores for all the semi-finals matches**



**Source: 538 by Nate Silver**

# Statistical modeling terminology

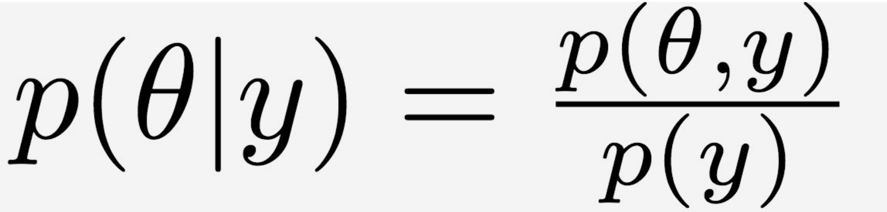
- $y$ - data
- $\theta$ - parameters
- $p(y, \theta)$ - ***joint*** probability distribution of the data and parameters
- $p(y \mid \theta)$ - ***conditional*** probability of the data given the parameters
  - if $y$ is fixed, this is the *likelihood function*
  - if $\theta$ is fixed, this is the *sampling function*
- $p(\theta \mid y)$ - ***posterior*** probability distribution

  the probability of the parameters given the data
- $p(\theta)$ - ***prior*** probability distribution

  the probability of the parameters before any data are observed
- $p(\tilde{y} \mid y)$ - ***posterior predictive*** distribution

  the probability of new data ($\tilde{y}$) conditioned on observed data ($y$)

# Bayes's Rule



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

Relates the ***posterior*** probability to the ***joint*** probability

$p(\theta \mid y) = p(\theta\,, y) \div p(y)$

$= (p(\theta) \times p(y \mid \theta)) \div p(y)$ [apply Bayes rule to numerator]

[drop denominator because factor $p(y)$ doesn't depend on $\theta$
and is constant for fixed $y$ ]

$\propto p(\theta) \times p(y \mid \theta)$ [unnormalized posterior density]

The ***posterior*** is proportional to the ***prior*** times the ***likelihood***

# *“quantifying uncertainty in inferences”*

$$p(\theta \mid y) \propto p(\theta) \times p(y \mid \theta)$$

The ***posterior*** is proportional to the ***prior*** times the ***likelihood***.

We can compute the mean, median, mode.

Quantiles of the posterior distribution provide **credible intervals.**

# Bayesian Workflow

Simple workflow:

- (Data gathering, preliminary data analysis)
- Build the full joint probability model -
  use everything you know about the world and the data
- Fit data to model (using Stan!)
- Evaluate the fit:
  - how well?
  - do the predictions make sense?
  - how sensitive are the results to the modeling assumptions?

# Stan

*the man, the language, the software*

Stanislaw Ulam - originator of Monte Carlo (MC) estimation techniques

Probabilistic programming language

Stan NUTS-HMC sampler - Markov Chain Monte Carlo (MCMC) sampler

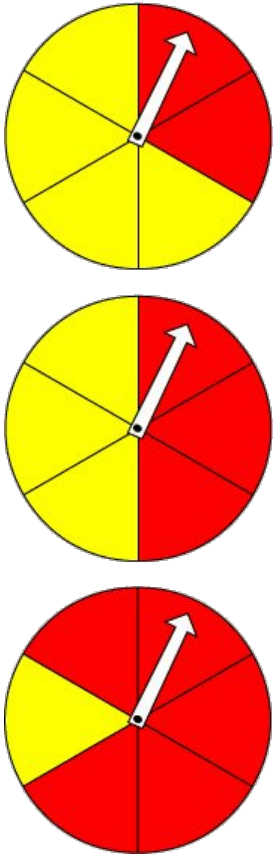
Rich eco-system of downstream analysis packages (but not enough in Python!)

Open-source - [https://github.com/stan-dev/stan](https://github.com/stan-dev/stan)

# Stan programming language example model:

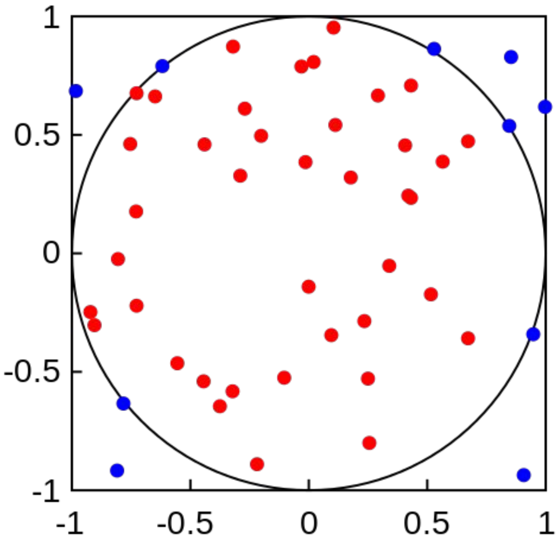
## bernoulli.stan

```
data {
  int<lower=0> N;
  int<lower=0,upper=1> y[N];
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ beta(1,1);
  y ~ bernoulli(theta);
}
```



# Monte Carlo Simulation: Calculate $\pi$

- Computing $\pi$ = 3.14... via simulation is *the* textbook application of Monte Carlo methods.
- Generate points (x,y) uniformly at random within range (-1, 1)
- Calculate proportion within unit circle: $x^2 + y^2 < 1$
- Area of a circle is $\pi\ r^2$, area of the unit circle is $\pi$
- Area of the square is 4
- Ratio of points inside the circle to total points is $\pi$ / 4
- $\pi$ = points inside circle × 4



# Monte Carlo Simulation: Calculate $\pi$

```
import numpy as np
def estimate_pi(n: int) -> float:
    xs = np.random.uniform(-1,1,n)
    ys = np.random.uniform(-1,1,n)
    dist_to_origin = [x**2 + y**2 for x,y in zip(xs, ys)]
    in_circle = sum(dist < 1 for dist in dist_to_origin)
    pi = float(4 * (in_circle / n))
    return pi
```

| n | pi | time elapsed (seconds) |
|---|---|---|
| 100 | 3.5 | 0.0008 |
| 10,000 | 3.15 | 0.03 |
| 1,000,000 | 3.139 | 3.2 |
| 100,000,000 | 3.1413 | 323.8 |

# Markov Chain Monte Carlo (MCMC)

- Standard MC estimation uses set of independent, identically distributed (i.i.d.) draws according to probability function *p(θ )*, e.g. `np.random.uniform(-1,1,n)`
- For complex Bayesian models, we can’t compute this function.
- A Markov Chain is a sequence of draws where the conditional probability of each draw depends only on the previous draw.
- This only happens once the Markov Chain has *converged*.
- *Warmup* is the process of getting to convergence.
- *If the chain has not converged, your sample is not valid.*

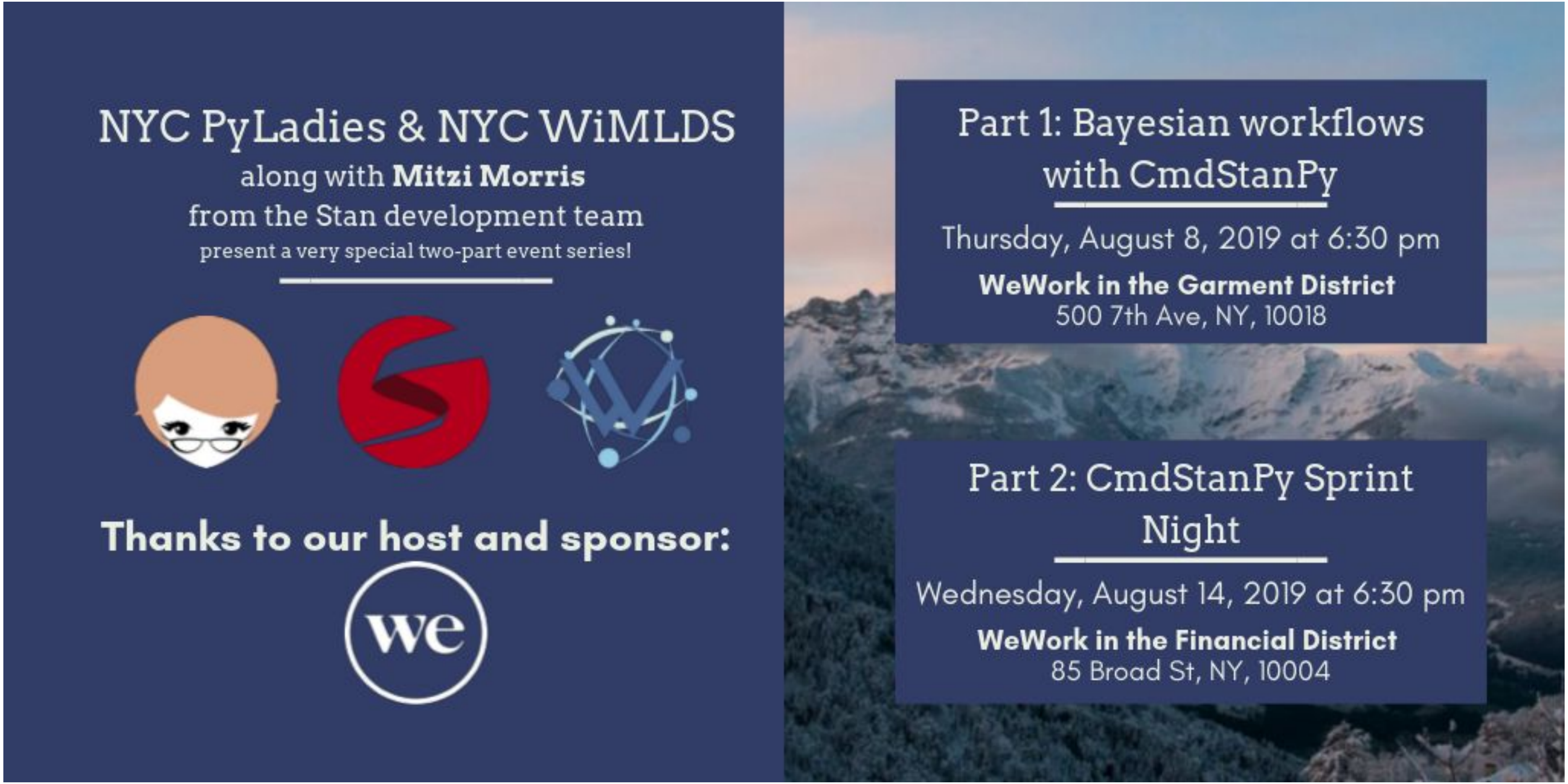
# Stan’s secret sauce:  HMC-NUTS sampler

- Algorithm for efficient MCMC sampling
- Not actually secret:  same algorithm used in PyMC3 and Edward
- References and tutorials:
    - Hoffman and Gelman, 2014
    - Betancourt videos
    - Stan User’s Guide

# CmdStanPy

- Designed to be lightweight
  - minimal package dependencies
  - minimal use of in-memory data structures
  - good for production workflows
- Keeps up with latest Stan release
- BSD license
- Requirements:
  - Python3
  - C++ (comes with anaconda or Xcode)

# Join us next week for the CmdStanPy Sprint Night!



NYC PyLadies & NYC WiMLDS

along with **Mitzi Morris**

from the Stan development team

present a very special two-part event series!

**Thanks to our host and sponsor:**

Part 1: Bayesian workflows with CmdStanPy

Thursday, August 8, 2019 at 6:30 pm

**WeWork in the Garment District**
500 7th Ave, NY, 10018

Part 2: CmdStanPy Sprint Night

Wednesday, August 14, 2019 at 6:30 pm

**WeWork in the Financial District**
85 Broad St, NY, 10004

# Live Demo: Stan goes to the Women's World Cup!

- **Colab: bit.ly/pyladiesstan3**
  Colab will be in "Playground mode" - i.e, you can run the notebook but not make any edits.
  If you would like to make edits (i.e., "personalize" it), make a copy onto your drive!

- **Github: bit.ly/CmdStanPyLadies**
  Follow the Readme instructions to get set up!

# Acknowledgements

WeWork

PyLadies

WiMLDS

Nitya!  Melissa!  Felice!

and the rest of the Stan Development team

# Resources and References

Stan Users Guide

- Models and programming techniques
- See: section on IRT models

Stan Reference Manual

- Stan language syntax and semantics
- Algorithms for sampling and estimation methods

Stan Discussion Groups on Discourse

Andrew Gelman blogpost on the 2014 World Cup model
(and the rest of Andrew's blog posts)