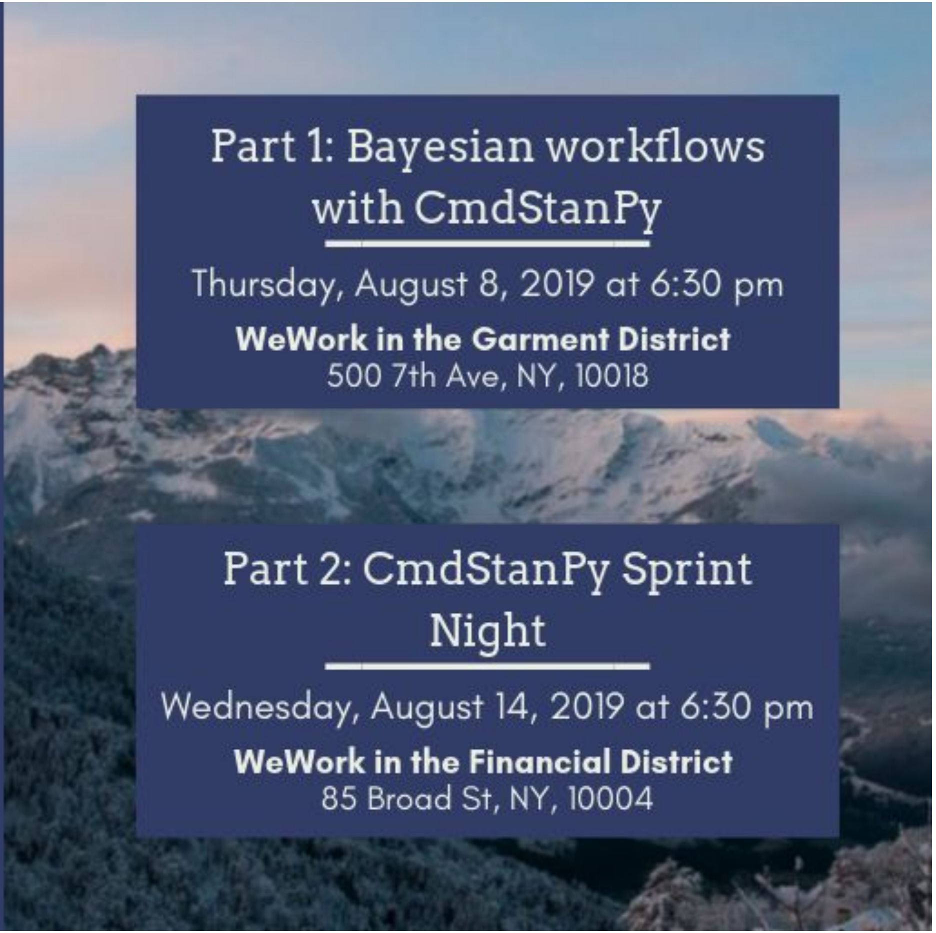
NYC PyLadies & NYC WiMLDS

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

pyladies

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.

WiMLDS
Women in Machine Learning & Data Science

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

sprint_details.md

What is Open Source

- Source code freely available to users
- Software made by many people
- Based on license
    - Can modify source and distribute own versions of program
        - Python - OSI-approved license
        - CmdStanPy - BSD License
- Sprints
    - Learn more about a package
    - Contribute to improvements
        - documentation, bug fixes, testing, feedback
    - Collaborate, build momentum and community

sprint_details.md

Contributing to Open Source

- Give back to community
- Improve quality of software, fewer bugs
- Encourages culture of collaboration
- Further develop as a programmer:
    - Improve documentation
    - Write clean, maintainable code

sprint_details.md

Getting Started

- Set up GitHub account (github.com)
- Requires using Git (git-scm.com)
    - github.com/reshamas/git-intro-workshop
- Review CmdStanPy README.md and documentation
    - https://cmdstanpy.readthedocs.io/en/latest/index.html
- Review issues page
    - http://bit.ly/cmdstanpy-sprint-0814
- Fork the repository
    - https://github.com/stan-dev/cmdstanpy

Sprint Night

- Curated issues list
  - http://bit.ly/cmdstanpy-sprint-0814
- Areas of focus:
  - Documentation
  - Beta testing (break it, log any issues)
  - Case studies - build jupyter notebooks
  - Performance testing (stress load with large datasets)
- Ask Mitzi Morris any questions on CmdStanPy
- Ask Felice or Nitya for help on issues getting started
  - Join Gitter https://gitter.im/nyc-pyladies/2019-cmdstanpy-bayesian-workshop

Session Outline

- Setup/Installation
- CmdStanPy Review / Under the Hood
- Tickets
- Join forces!

Bayesian Workflow - model specification

- Data gathering, preliminary data analysis
  ‘y’ is the observed outcome, ‘x’ are the inputs

- Build the full joint probability model, i.e. write a Stan program

- Compile the program:

```
my_model = Model(stan_file=os.path.join(‘path’, ‘my_model.stan’))
my_model.compile()
```

- This might take a few tries...

  (demo notebook ‘Workflow - examples of syntax errors’)

Bayesian Workflow - fit model to data

- Create an in-memory Dict or json file containing a single object which has entries corresponding to all variables declared in the program’s data block.

```
my_data = { 'N': 10, 'y': [0,0,1,0,1,1,1,0,0,0] }
```

- Fit data to model:

```
my_fit = my_model.sample(data=my_dat›, chains=4)
```

- Check the fit:

```
my_fit.diagnose()
my_fit.summary()
```

Bayesian Workflow - model evaluation

- Get the posterior sample (or estimate)

```
my_drawset = my_fit.sample()
```

- **Create visualizations -** this is actually outside of CmdStanPy

  but it’s what matters (like data collection and prelimiary data analysis)

  for this workshop, compling recipes for visualization

  and exploring Python packages (arviz?) would be great!

Resources

Stan Users Guide

- Models and programming techniques

Stan Reference Manual

- Stan language syntax and semantics

Stan Language Functions Reference

- Probability distributions and math functions available in the Stan language

Under the hood

- CmdStanPy uses Python’s subprocess library to call CmdStan

- CmdStan is file-based

- CmdStanPy creates per-session temporary directory
  - output files will be written to this directory by default
  - StanFit object provides methods to assemble output into in-memory numpy.ndarray or pandas.DataFrame
    or
    move/rename set of output files to permanent location