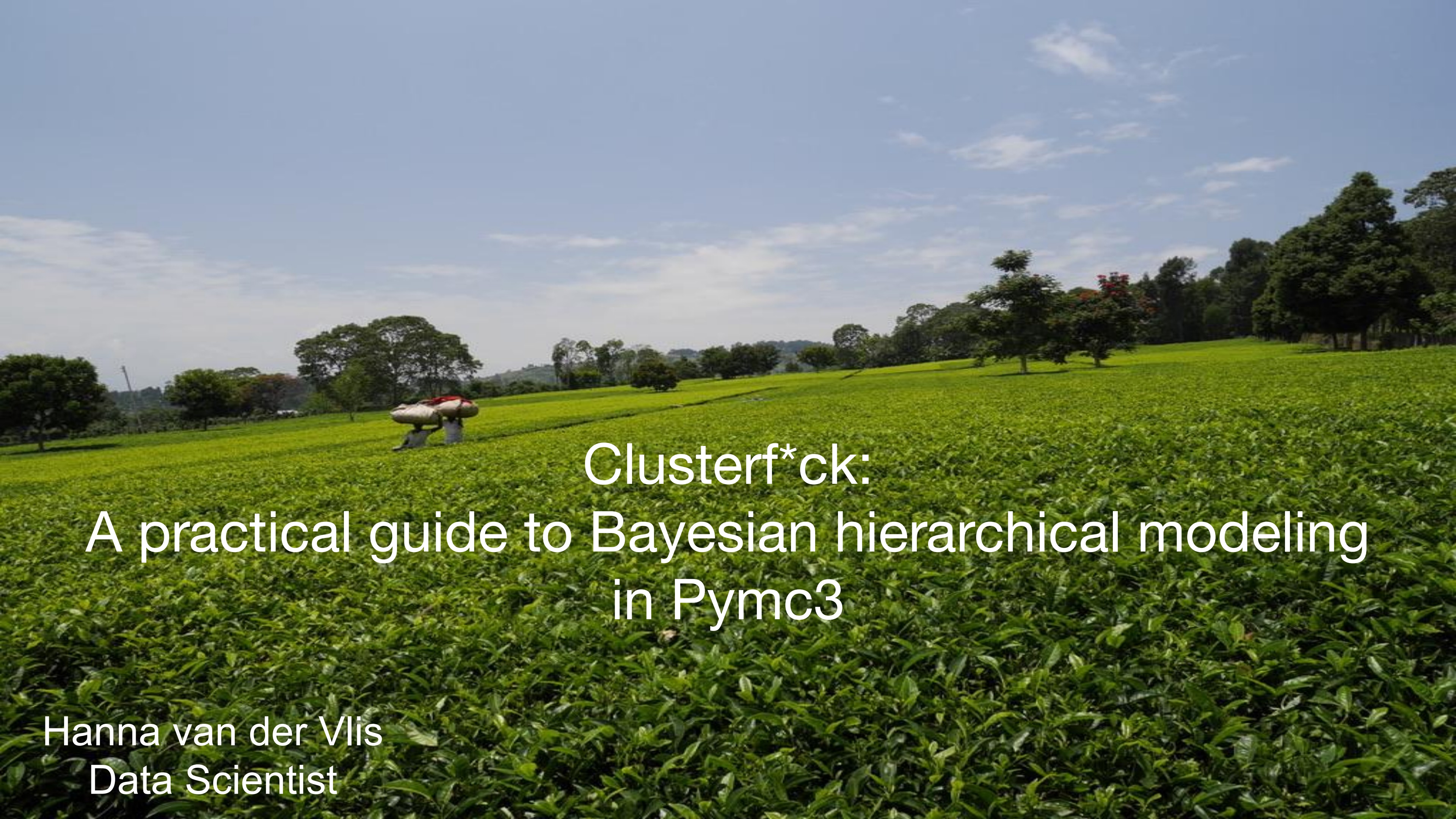


Clusterf*ck: A practical guide to Bayesian hierarchical modeling in Pymc3

Hanna van der Vlis
Data Scientist

Introduction to Apollo Agriculture

Fintech / agritech

Mission Helping smallholder farmers become more profitable, using:

- Access to capital
- Good quality farm inputs
- Training
- Insurance
- Market access

Real-world example: predicting yield

Predictor variables:

- Seed & fertilizer used
- Prior experience with farming
- Demographic data
- County

Target variable:

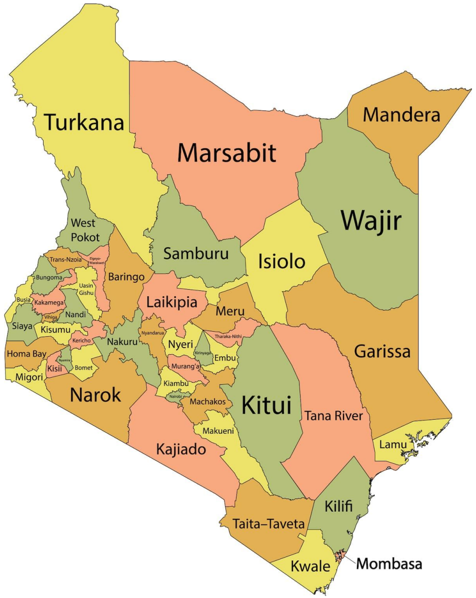
- Average bags of maize per acre

However….

We see **regional clusters** of high yields in a given season.

These clusters are not, however, consistent year over year.

→ even if we exclude county/region as a variable in the model



How do we address hierarchical data?

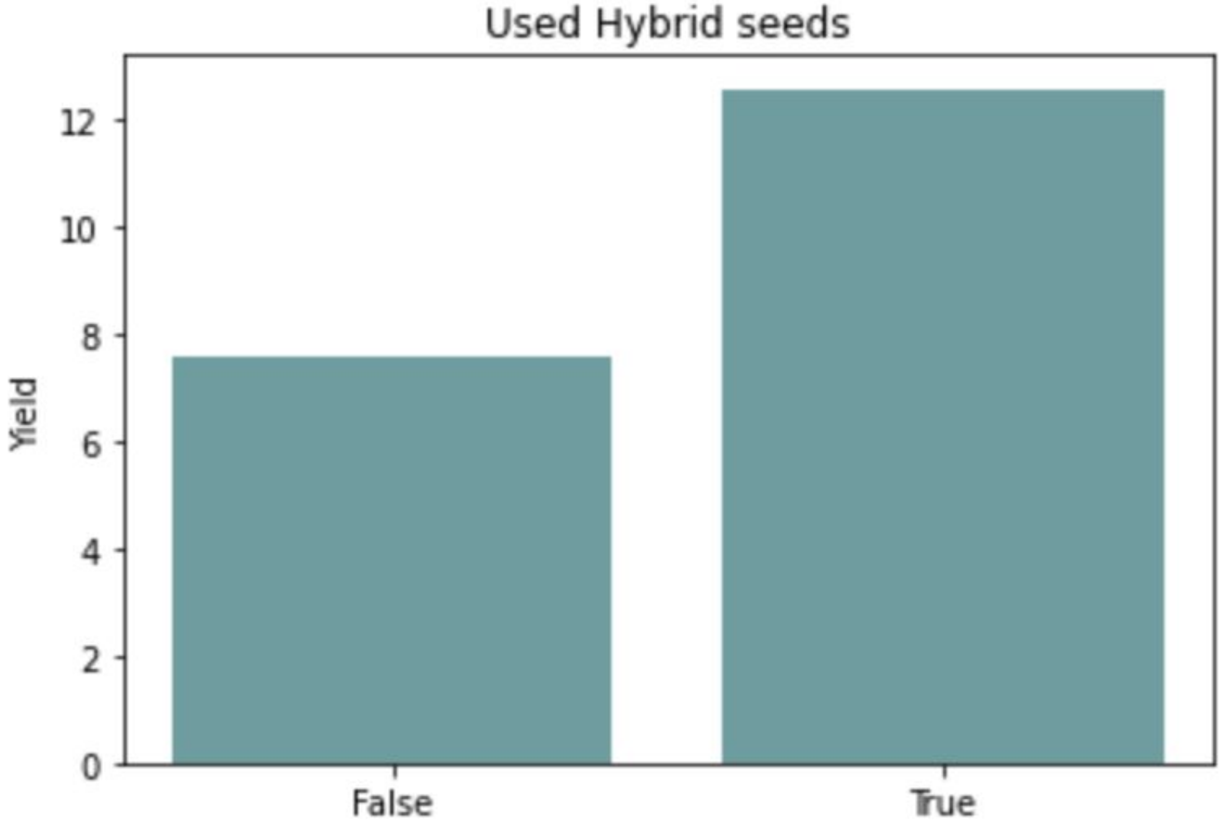
- Pooled model - Ignore the grouping structure
  - Important information is neglected
- Unpooled - Model each group separately
  - Many distinct groups
  - Some groups have small sample sizes
  - Not using all data maximally
- Hierarchical model - mixing of the two
  - Complexity

Use-case with real world data

	county	maize_yield	used_hybrid_seeds
0	Baringo	12.00	True
1	Baringo	18.75	True
2	Baringo	20.00	True
3	Baringo	10.00	True
4	Baringo	10.00	True
...	...	...	...
110784	West Pokot	7.00	True
110785	West Pokot	33.00	True
110786	West Pokot	4.00	True
110787	West Pokot	25.00	True
110788	West Pokot	10.00	True

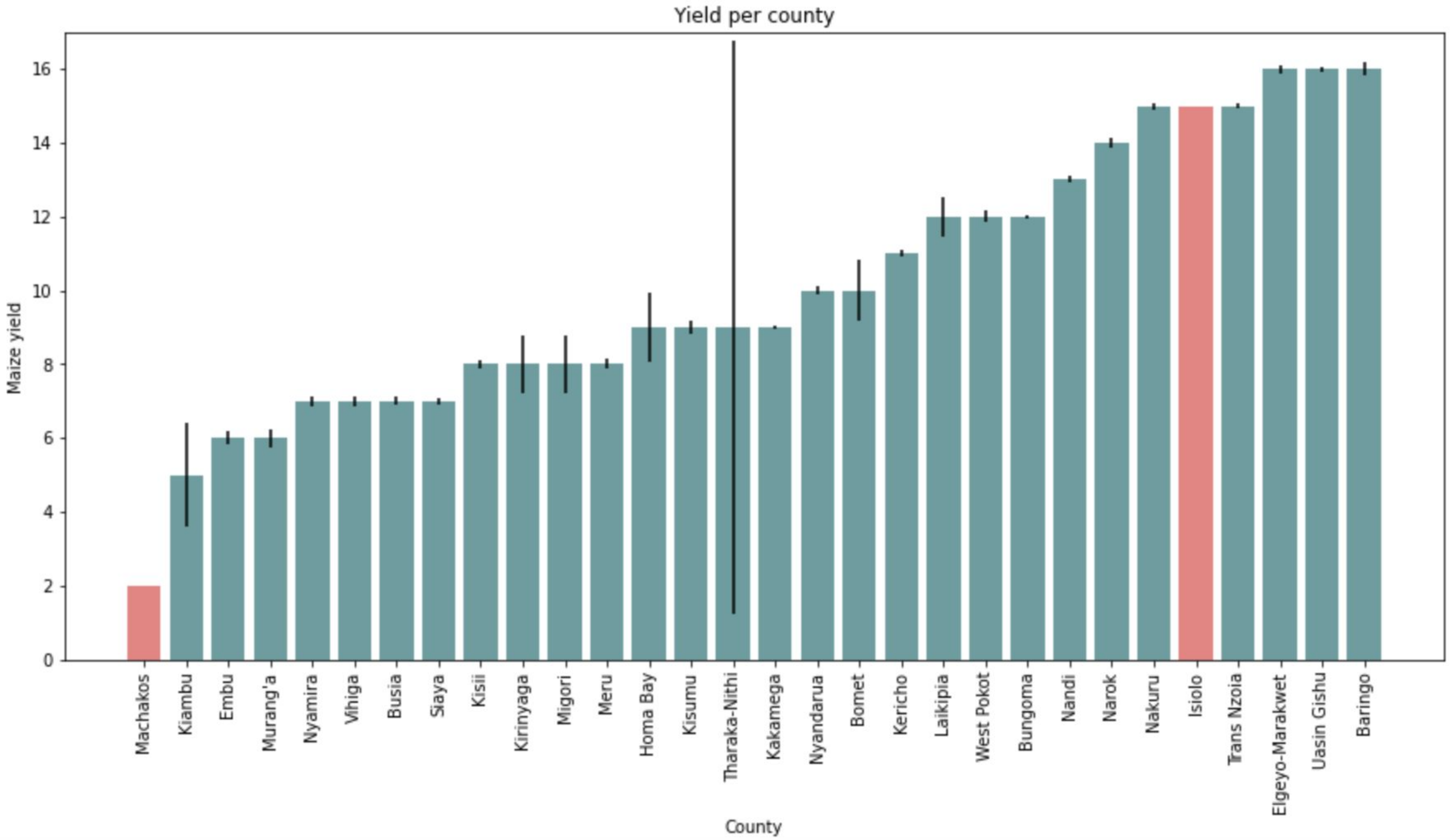
110789 rows × 3 columns

Difference in yield when using hybrid seeds



= only 1 observation

Differences in yield between counties



Bayesian framework

Why?

- Allows you to incorporate prior knowledge
- Model the outcome in terms of a probability distribution - direct quantification of uncertainty

Bayesian data analysis - an overview

1. Setting up a full probability model
2. Calculate and interpret the appropriate posterior distribution conditional on the observed data
3. Evaluate model fit

– repeat –

PyMC3

PyMC3 allows you to write down models using an intuitive syntax to describe a data generating process

```python
import pymc3 as pm
```

Prerequisites

```
counties = data.county.unique() # list of length 30
n_counties = len(counties) # int(30)
county_lookup = dict(zip(counties, range(n_counties))) # dict with county <> integer mapping 0-29
```

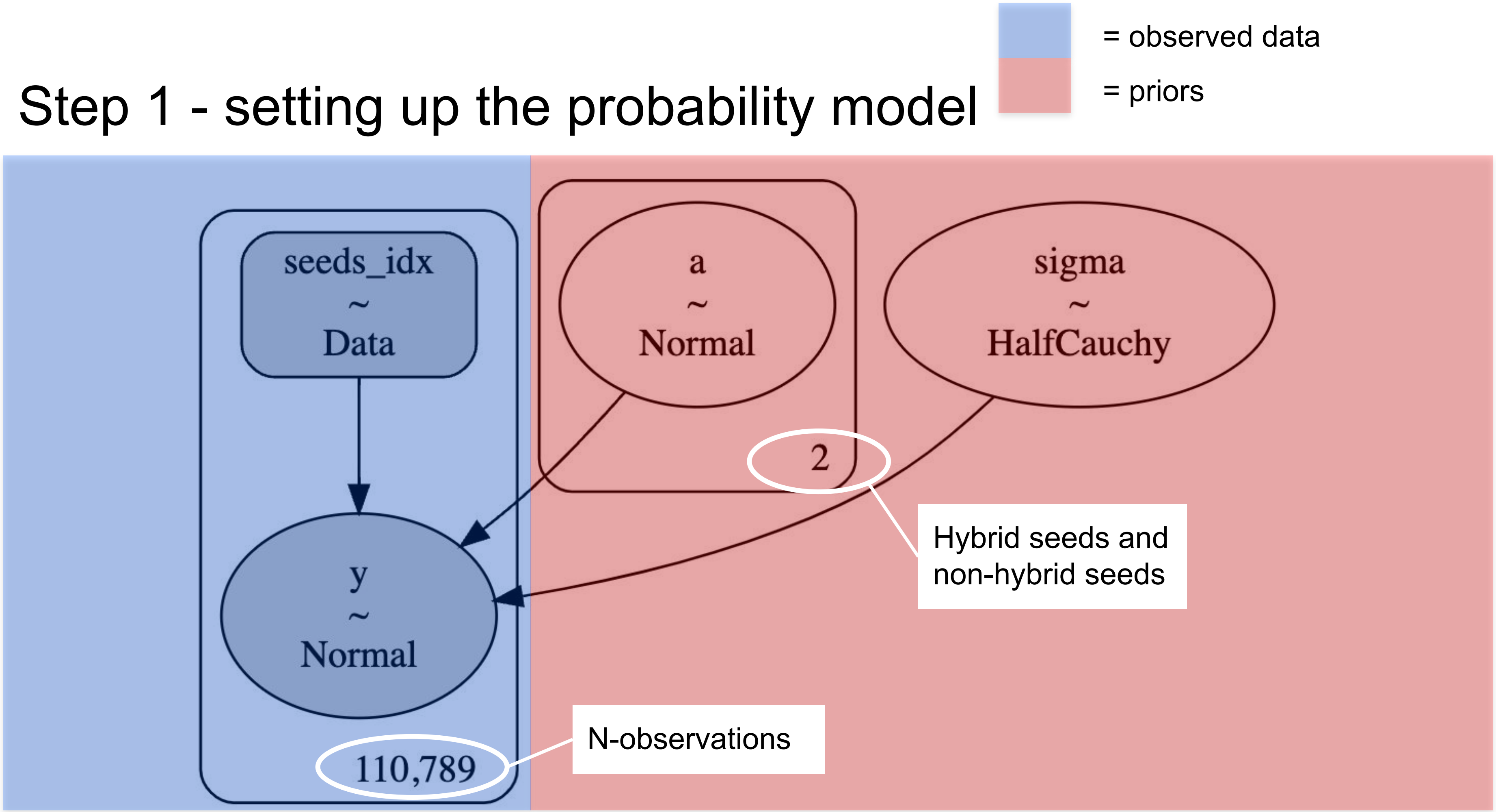
```
county = data["county_code"] = data.county.replace(county_lookup).values # array of len 110789 with county code
maize_yield = data['maize_yield'] # series of length 110789 with maize yield: float
hybrid_seeds = data.used_hybrid_seeds.astype(int).values # array of length 110789: 0 for non-hybrid and 1 for hybrid
```


Pooled model

Code example

```
coords = {"Hybrid_seeds": ["False", "True"], "obs_id": np.arange(hybrid_seeds.size)}
with pm.Model(coords=coords) as pooled_model:
    seeds_idx = pm.Data("seeds_idx", hybrid_seeds, dims="obs_id")
    a = pm.Normal("a", mu=2.3, sigma=1.1, dims="Hybrid_seeds")

    theta = a[seeds_idx]
    sigma = pm.HalfCauchy("sigma", 1)

    y = pm.Normal("y", mu=theta, sigma=sigma, observed=log_maize_yield, dims="obs_id")

pm.model_to_graphviz(pooled_model)
```

Step 1 - setting up the probability model



Choosing distributions

```
coords = {"Hybrid_seeds": ["False", "True"], "obs_id": np.arange(hybrid_seeds.size)}
with pm.Model(coords=coords) as pooled_model:
    seeds_idx = pm.Data("seeds_idx", hybrid_seeds, dims="obs_id")
    a = pm.Normal("a", mu=2.3, sigma=1.1, dims="Hybrid_seeds")

    theta = a[seeds_idx]
    sigma = pm.HalfCauchy("sigma", 1)

    y = pm.Normal("y", mu=theta, sigma=sigma, observed=log_maize_yield, dims="obs_id")

pm.model_to_graphviz(pooled_model)
```

Choosing distributions

- WTF is a conjugate prior?
  - Just remember: Gaussian x Gaussian = Gaussian! :)

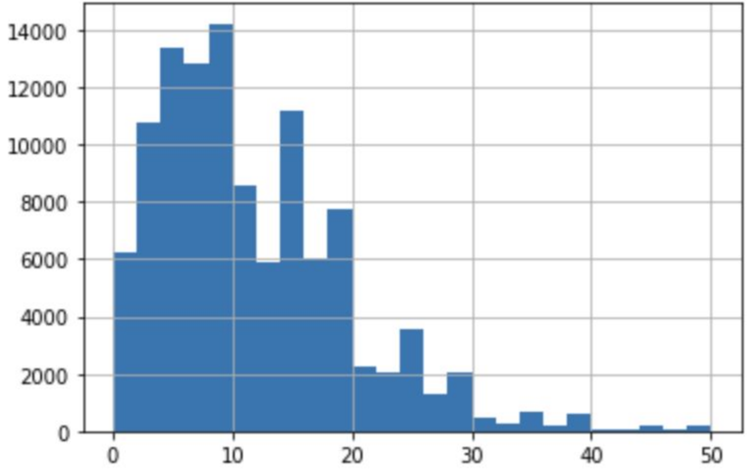
Can anyone explain conjugate priors in simplest possible terms?

Asked 6 years, 8 months ago   Modified 1 year, 10 months ago   Viewed 7k times

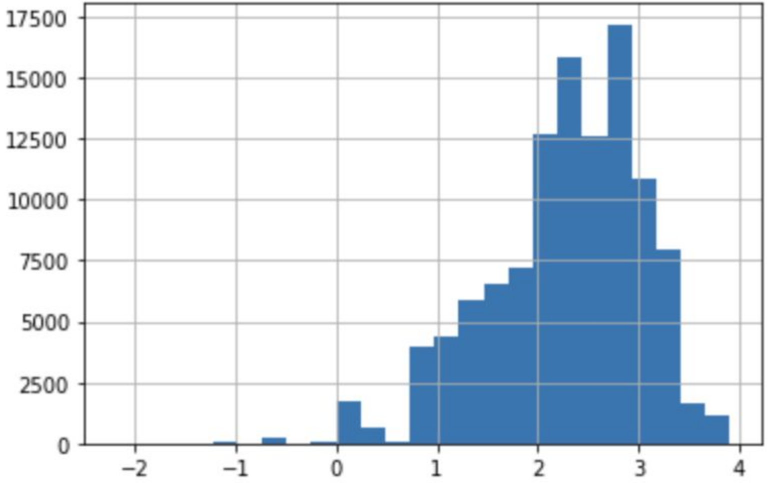
…or read this!

Data transformations

```
data.maize_yield.hist(bins=25);
```




```
data.log_maize_yield.hist(bins=25);
```




```
data["log_maize_yield"] = log_maize_yield = np.log(maize_yield + 0.1).values
```

Setting priors

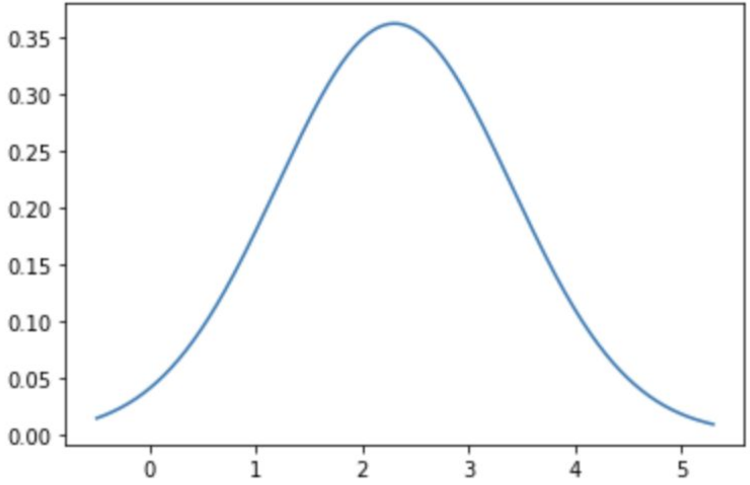
Express our knowledge (and uncertainty) about θ using domain knowledge or previous research.

The prior distribution should include all plausible values of θ, but the distribution need not be realistically concentrated around the true value

Setting priors

Normal distribution with *mu*=10

`np.log(10)=2.3` and `np.log(3)=1.1`

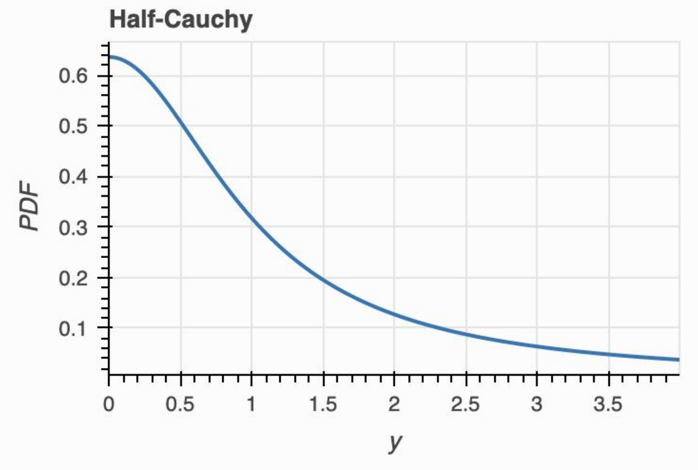



```
coords = {"Hybrid_seeds": ["False", "True"], "obs_id": np.arange(hybrid_seeds.size)}
with pm.Model(coords=coords) as pooled_model:
    seeds_idx = pm.Data("seeds_idx", hybrid_seeds, dims="obs_id")
    a = pm.Normal("a", mu=2.3, sigma=1.1, dims="Hybrid_seeds")

    theta = a[seeds_idx]
    sigma = pm.HalfCauchy("sigma", 1)

    y = pm.Normal("y", mu=theta, sigma=sigma, observed=log_maize_yield, dims="obs_id")

pm.model_to_graphviz(pooled_model)
```

Setting priors

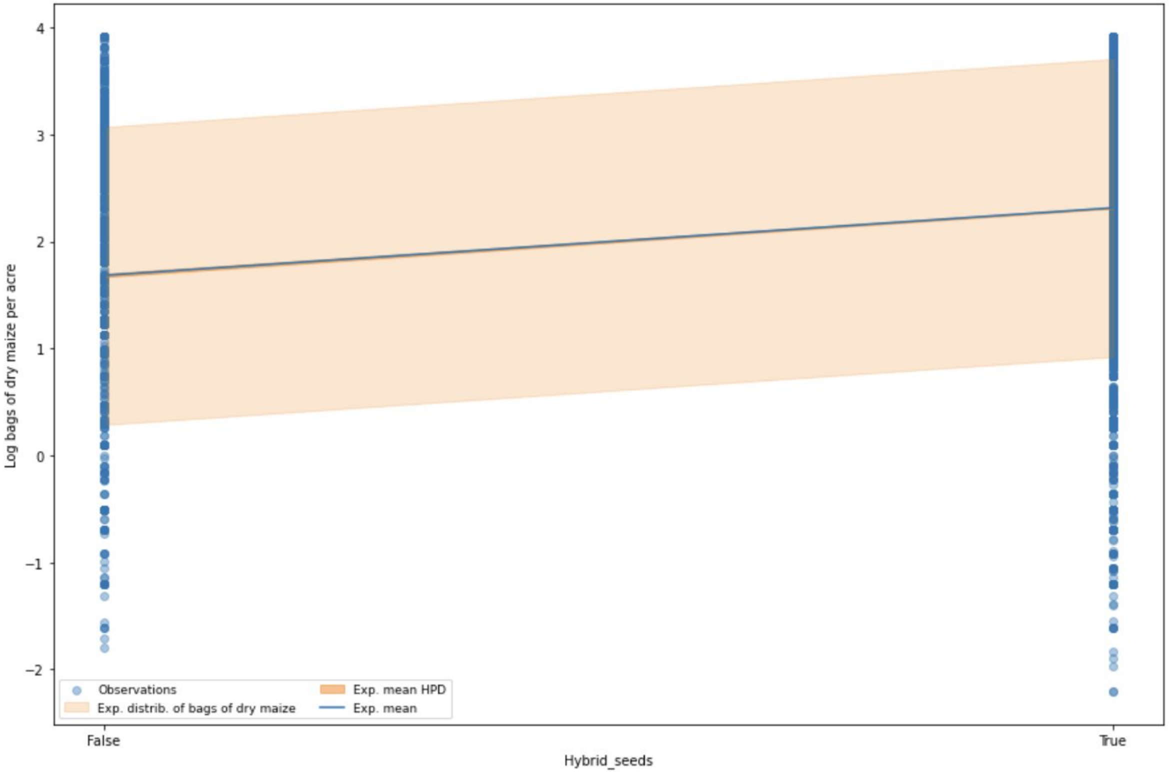
Weakly informative prior

- Broad peak at zero, and scale parameter 1



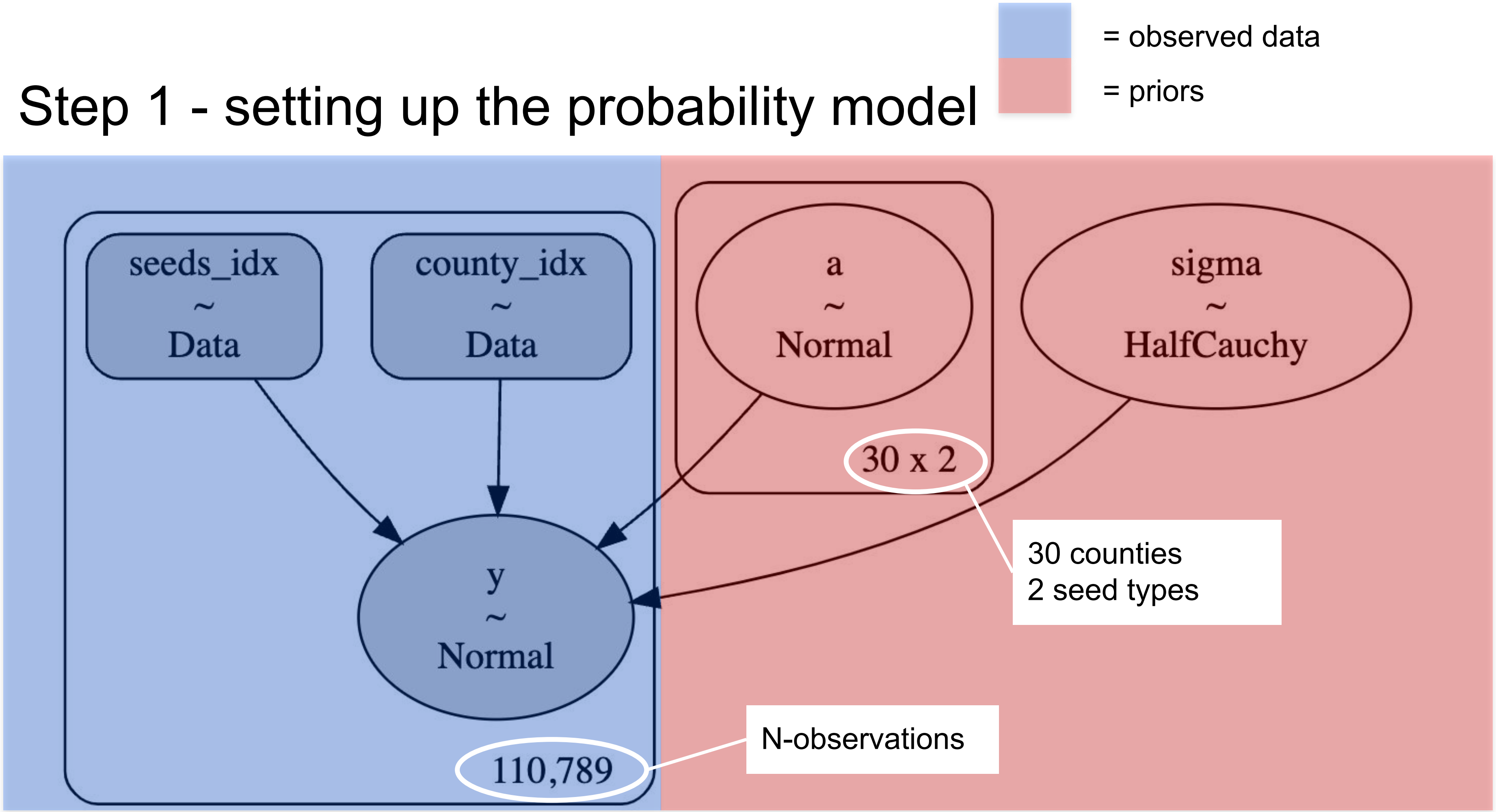

```
coords = {"Hybrid_seeds": ["False", "True"], "obs_id": np.arange(hybrid_seeds.size)}
with pm.Model(coords=coords) as pooled_model:
    seeds_idx = pm.Data("seeds_idx", hybrid_seeds, dims="obs_id")
    a = pm.Normal("a", mu=2.3, sigma=1.1, dims="Hybrid_seeds")

    theta = a[seeds_idx]
    sigma = pm.HalfCauchy("sigma", 1)

    y = pm.Normal("y", mu=theta, sigma=sigma, observed=log_maize_yield, dims="obs_id")

pm.model_to_graphviz(pooled_model)
```

Step 2 & 3 - interpret the posterior and evaluate model fit




Unpooled model

Step 1 - setting up the probability model



```python
coords["County"] = counties
with pm.Model(coords=coords) as unpooled_model:
    seeds_idx = pm.Data("seeds_idx", hybrid_seeds, dims="obs_id")
    county_idx = pm.Data("county_idx", county, dims="obs_id")
    a = pm.Normal("a", 2.3, sigma=1.1, dims=("County", "Hybrid_seeds"))

    theta = a[county_idx, seeds_idx]
    sigma = pm.HalfCauchy("sigma", 1)

    y = pm.Normal("y", theta, sigma=sigma, observed=log_maize_yield, dims="obs_id")
pm.model_to_graphviz(unpooled_model)
```

Step 2 & 3 - interpret the posterior and evaluate model fit

- **There are counties where yield with hybrid seeds is lower than without hybrid seeds (Machakos)**
- There are counties where the difference in yield between hybrid and non-hybrid is really high (Tharaka-Nithi)
- Many estimates have a lot of uncertainty due to small sample size

Step 2 & 3 - interpret the posterior and evaluate model fit

- There are counties where yield with hybrid seeds is lower than without hybrid seeds (Machakos)
- **There are counties where the difference in yield between hybrid and non-hybrid is really high (Tharaka-Nithi)**
- Many estimates have a lot of uncertainty due to small sample size

Step 2 & 3 - interpret the posterior and evaluate model fit

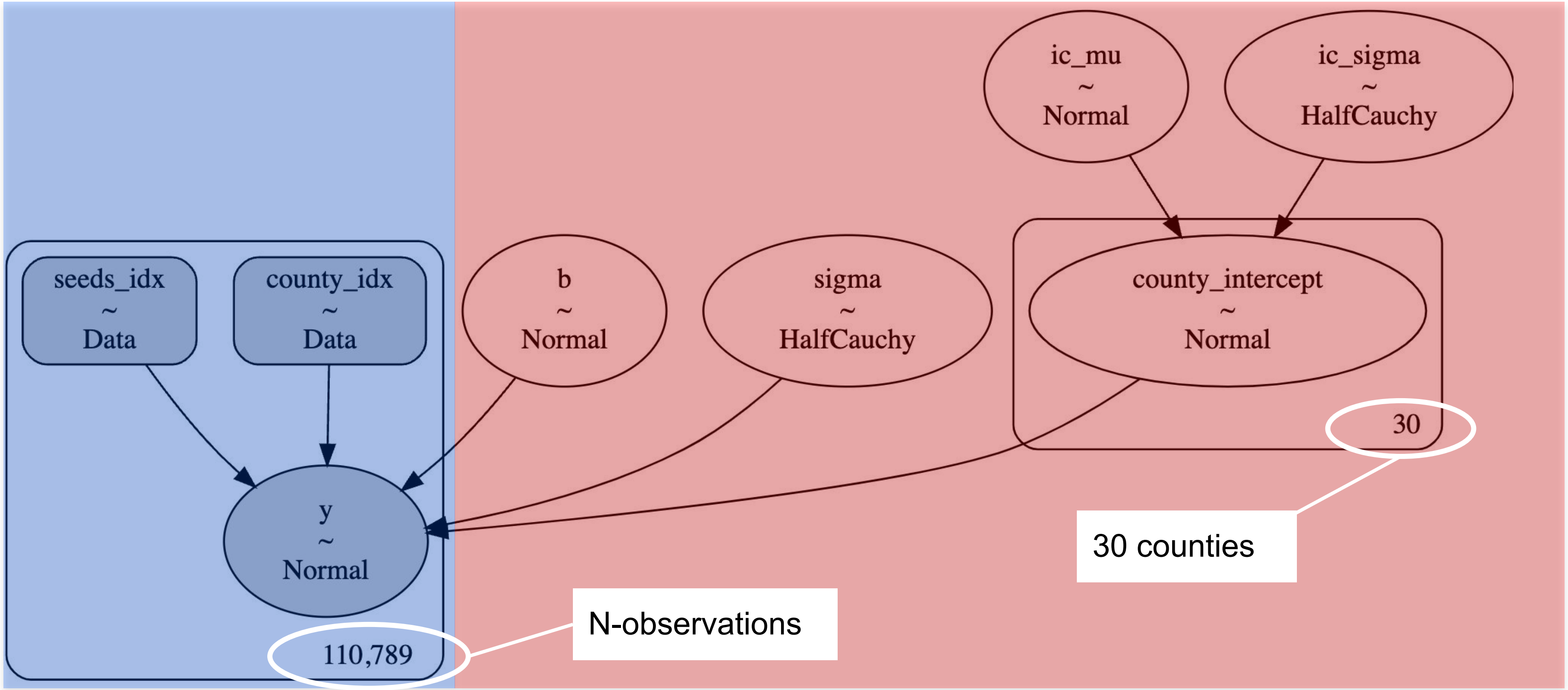
- There are counties where yield with hybrid seeds is lower than without hybrid seeds (Machakos)
- There are counties where the difference in yield between hybrid and non-hybrid is really high (Tharaka-Nithi)
- **Many estimates have a lot of uncertainty due to small sample size**

Hierarchical model

Step 1 - setting up the full probability model

= observed data

= priors



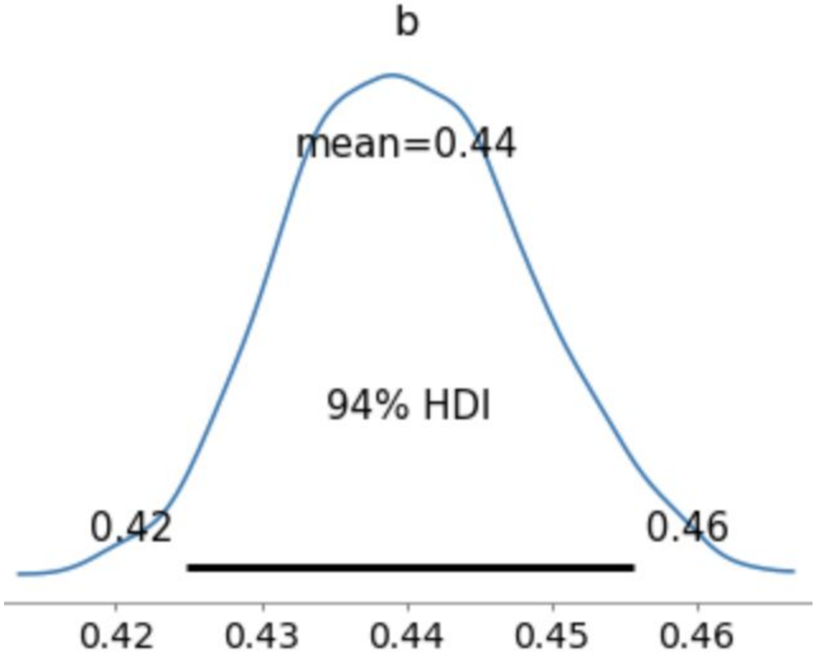
```python
with pm.Model(coords=coords) as varying_intercept:
    seeds_idx = pm.Data("seeds_idx", hybrid_seeds, dims="obs_id")
    county_idx = pm.Data("county_idx", county, dims="obs_id")
    # Hyperpriors:
    ic_mu = pm.Normal("ic_mu", mu=2, sigma=1)
    ic_sigma = pm.HalfCauchy("ic_sigma", 1.0)

    # Varying intercepts:
    county_intercept = pm.Normal("county_intercept", mu=ic_mu, sigma=ic_sigma, dims="County")
    # Common slope:
    b = pm.Normal("b", mu=2, sigma=1)

    # Expected value per county:
    theta = county_intercept[county_idx] + b * seeds_idx
    # Model error:
    sigma = pm.HalfCauchy("sigma", 1.0)

    y = pm.Normal("y", theta, sigma=sigma, observed=log_maize_yield, dims="obs_id")
pm.model_to_graphviz(varying_intercept)
```

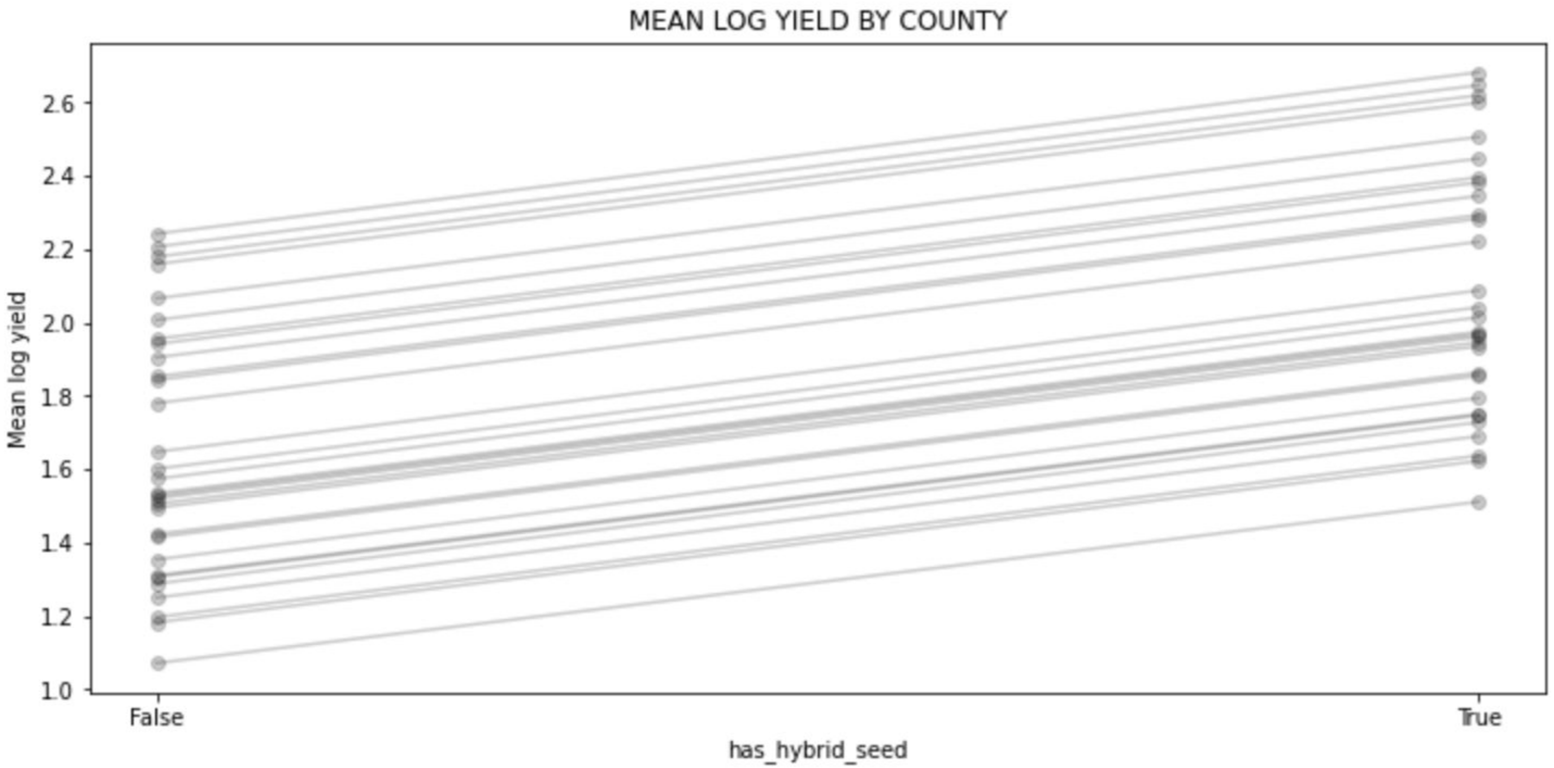
```
with pm.Model(coords=coords) as varying_intercept:
    seeds_idx = pm.Data("seeds_idx", hybrid_seeds, dims="obs_id")
    county_idx = pm.Data("county_idx", county, dims="obs_id")
    # Hyperpriors:
    ic_mu = pm.Normal("ic_mu", mu=2, sigma=1)
    ic_sigma = pm.HalfCauchy("ic_sigma", 1.0)

    # Varying intercepts:
    county_intercept = pm.Normal("county_intercept", mu=ic_mu, sigma=ic_sigma, dims="County")
    # Common slope:
    b = pm.Normal("b", mu=2, sigma=1)

    # Expected value per county:
    theta = county_intercept[county_idx] + b * seeds_idx
    # Model error:
    sigma = pm.HalfCauchy("sigma", 1.0)

    y = pm.Normal("y", theta, sigma=sigma, observed=log_maize_yield, dims="obs_id")
pm.model_to_graphviz(varying_intercept)
```

Step 2 - interpret the posterior distribution

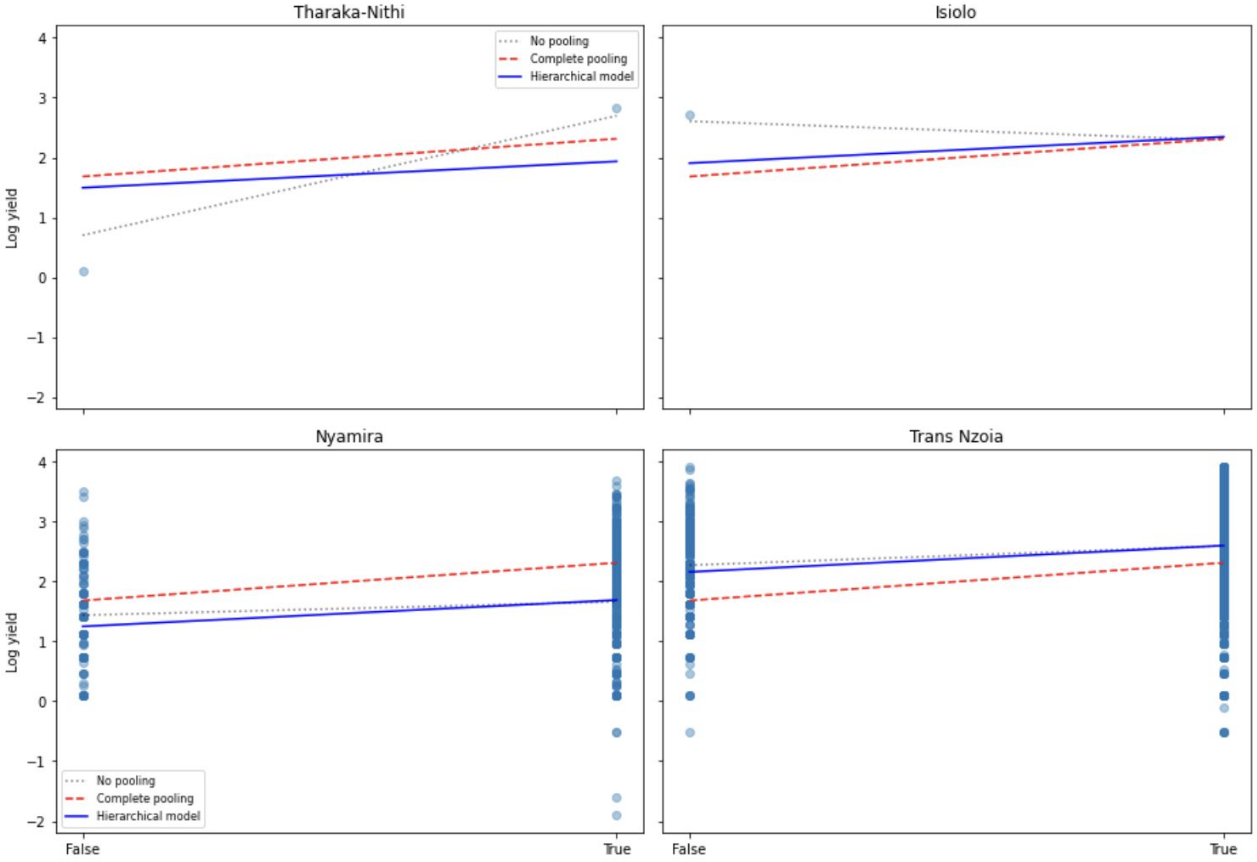
Farmers with hybrid seeds have about 1.6x (exp(0.44)) the yield of those without hybrid seeds, after accounting for county.

This is a *relative* effect.




MEAN LOG YIELD BY COUNTY
Mean log yield
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
False
True
has_hybrid_seed

Comparison of the three methods



What else can we do?

- Varying slopes
- Adding correlation between intercepts and slopes
- Adding more predictor variables
- Adding a hierarchical layer for year/season

References

This question on stackexchange that could’ve come from me:

https://stats.stackexchange.com/questions/176668/can-anyone-explain-conjugate-priors-in-simplest-possible-terms

Example blogposts:

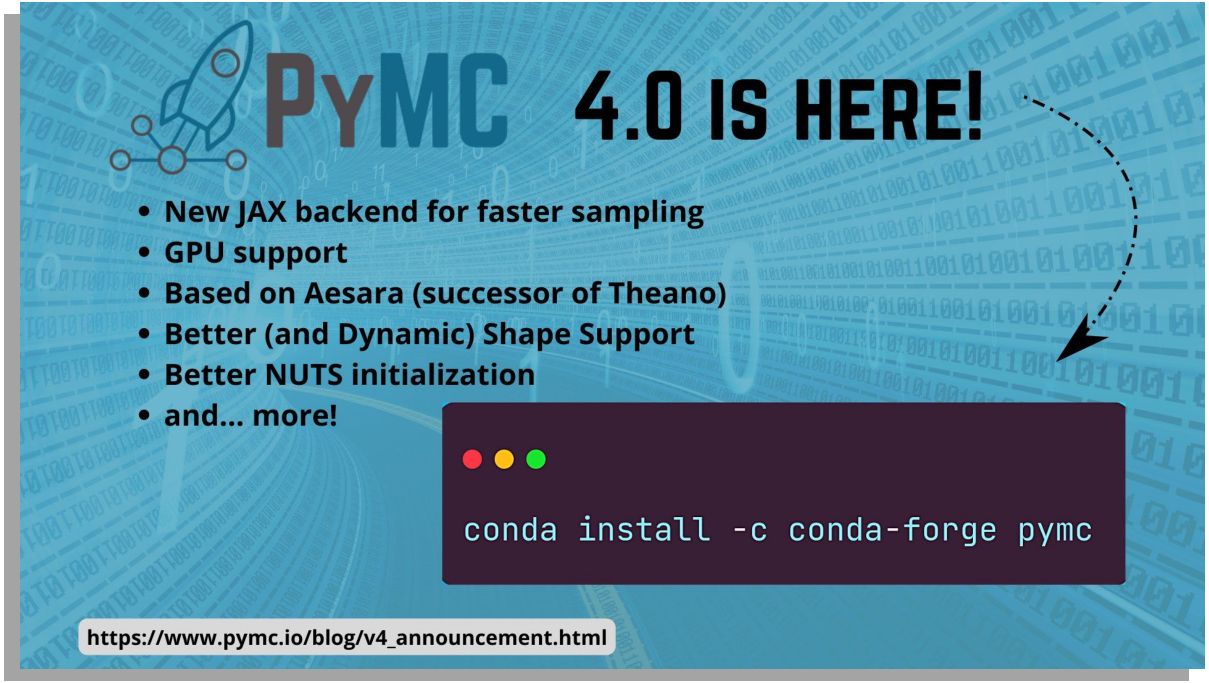
https://docs.pymc.io/en/v3/pymc-examples/examples/case_studies/multilevel_modeling.html#:~:text=A%20hierarchical%20model%20is%20a,but%20overlapping%2C%20clusters%20of%20parameters.

https://twiecki.io/

Trainings:

https://www.intuitivebayes.com/ – haven’t actually done this, just heard of it this weekend

Questions?



APOLLO AGRICULTURE

P.S.: we're hiring!

hanna@apolloagriculture.com

https://twitter.com/hannavdvlis

Now, how do you recognize clustering in your data?

- Think about how your data is sampled
- After modeling: plot acceptance rates or probability distributions over groups in which you sampled

When is clustered data important?

- If the pooled model gives you a biased generalization error
  - Biased data leads to biased empirical error
- If you want to measure to what extent different groups are pooled vs unpooled

Make explicit what is otherwise implicit!