

# Hierarchical Linear Models with PyMC3

---

Paul Black - Junior Data Scientist DRIVIN  
Metis - Chicago -2018

[Paul.laifu.black@gmail.com](mailto:Paul.laifu.black@gmail.com)  
[linkedin.com/in/paulfblack](https://www.linkedin.com/in/paulfblack)  
[github.com/paulfblack](https://github.com/paulfblack)



# DRIVIN

A KAR Auction Services Company



# Hierarchical Modeling

---

Hierarchical modeling is a tool of probabilistic programming and Bayesian statistics which predicts the parameters of a posterior distribution for data which can be observed on multiple levels.

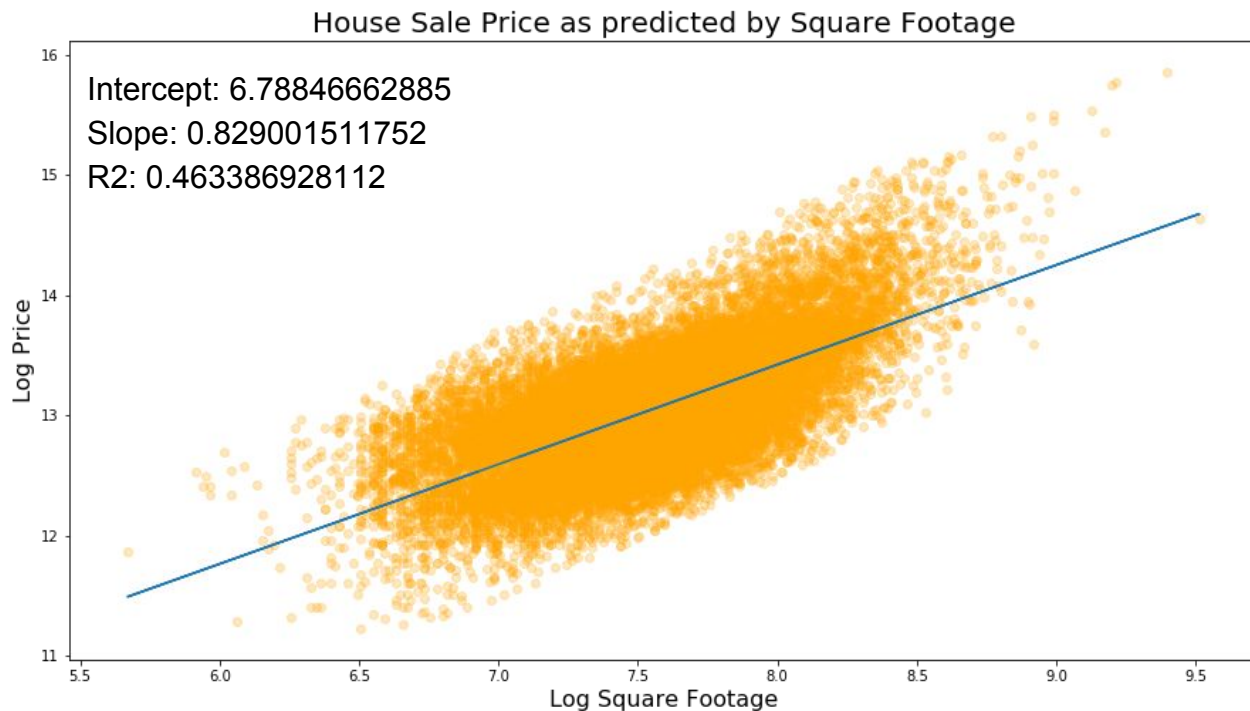
It allows for between group varying slope and varying intercept bounding the parameters between complete and no pooled alternatives.



# Case Study: Historical Housing Sale Data in Seattle

$$y = a + bx$$

	price	sqft_living	zipcode
0	221900	1180	98178
1	538000	2570	98125
2	180000	770	98028
3	604000	1960	98136
4	510000	1680	98074
5	1225000	5420	98053
6	257500	1715	98003
7	291850	1060	98198
8	229500	1780	98146
9	323000	1890	98038
10	662500	3560	98007
11	468000	1160	98115



Data source:

<https://www.coursera.org/learn/ml-foundations/supplement/RP8te/reading-predicting-house-prices-assignment>

# Case Study: Historical Housing Sale Data in Seattle

What are the three most important things in real estate?

- Location
- Location
- Location

By creating zip code indicators we can move from a single featured line:

$$y = a + bx$$

to

$$y = a + b_1x_1 + b_ix_i$$

	price	sqft_living	zipcode_98002	zipcode_98003	zipcode_98004	zipcode_98005	zipcode_98006
0	201000	900	1	0	0	0	0
1	300000	1984	1	0	0	0	0
2	142500	690	1	0	0	0	0
3	125000	920	1	0	0	0	0
4	213500	1220	1	0	0	0	0

Data source:

<https://www.coursera.org/learn/ml-foundations/supplement/RP8te/reading-predicting-house-prices-assignment>

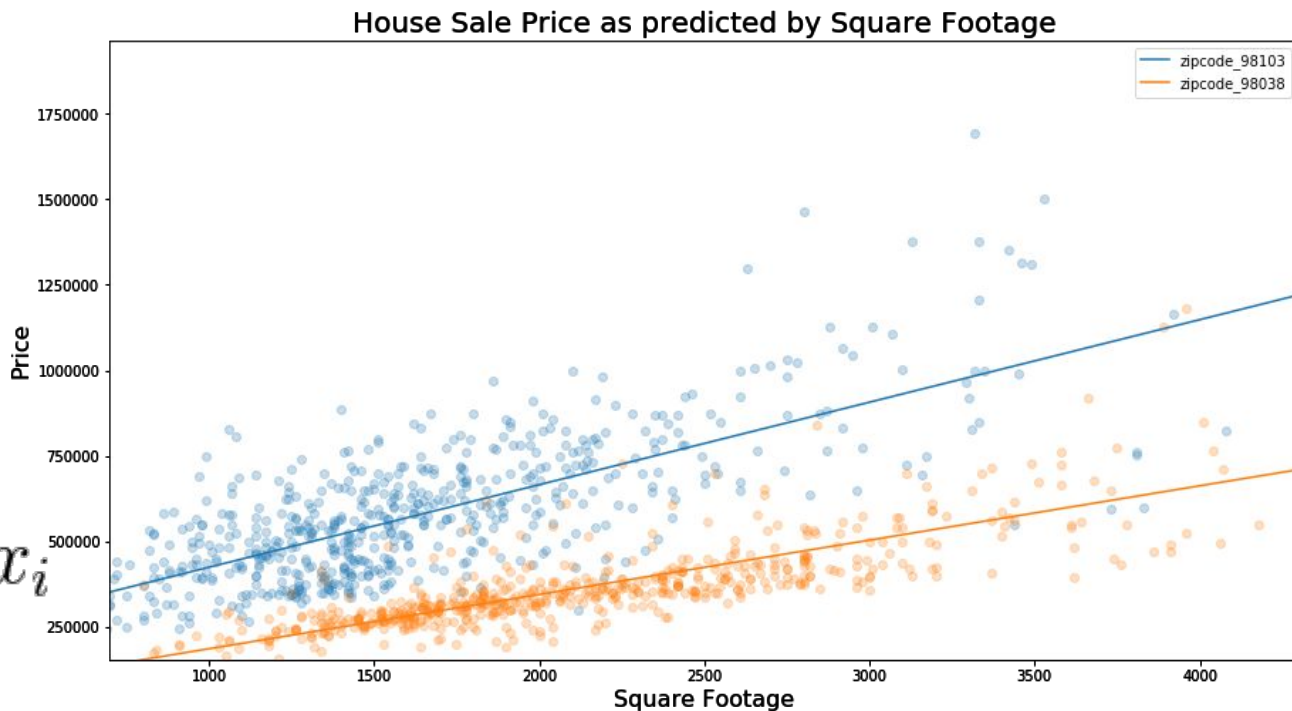
# Case Study: Historical Housing Sale Data in Seattle

## Varying Intercept

$$y = a + b_1x_1 + b_ix_i$$

## Varying Intercept Varying Slope

$$y = a + b_1x_1 + b_{n-1}x_ix_1 + b_nx_i$$



Data source:

<https://www.coursera.org/learn/ml-foundations/supplement/RP8te/reading-predicting-house-prices-assignment>

## Pooling: Complete, no, and partial pooling

---

In the varying-slope varying-intercept, what would happen if we predicted on a new zip code?

$$y = a + b_1x_1 + b_2x_2 + b_3x_1x_2 + \dots b_{n-1}x_ix_1 + b_nx_i$$

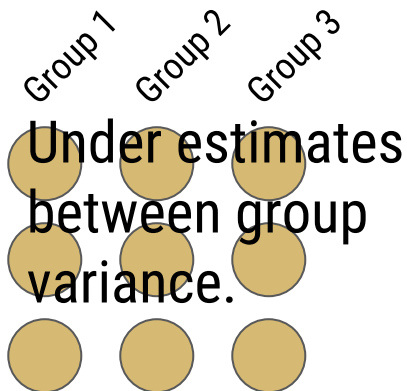
What if we had a zip code that lacked fully representative data? I.e. imagine we were predicting the effect of waterfront on price, but had coastal zip codes without waterfront property records

# Pooling: Complete, no, and partial pooling

---

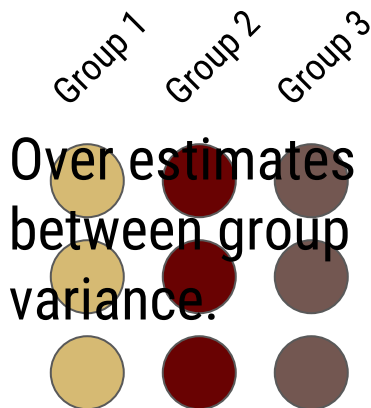
## Complete Pooling

All groups are given the same slope and intercept.



## No Pooling

All groups are allowed to have unique slopes and intercepts.



## Partial Pooling

The slopes and intercepts are related, but allowed to vary. They are assumed to come from a distribution of betas.

Bound between the extremes of complete and no pooling, pulling betas towards the mean.

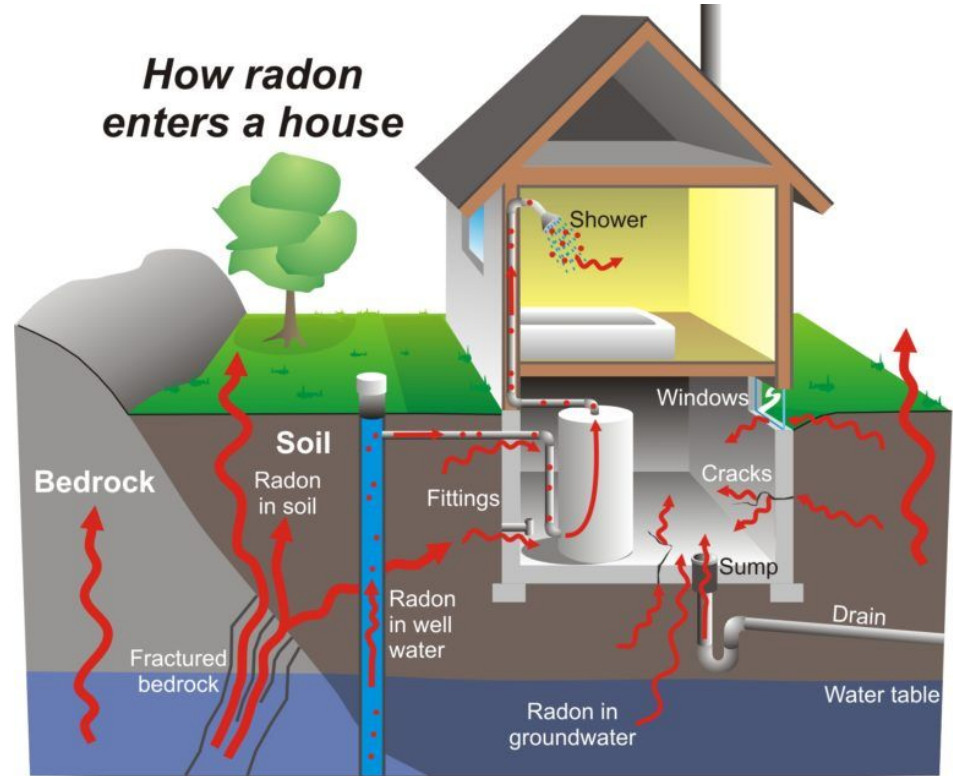


## Case Study: Radon Prediction

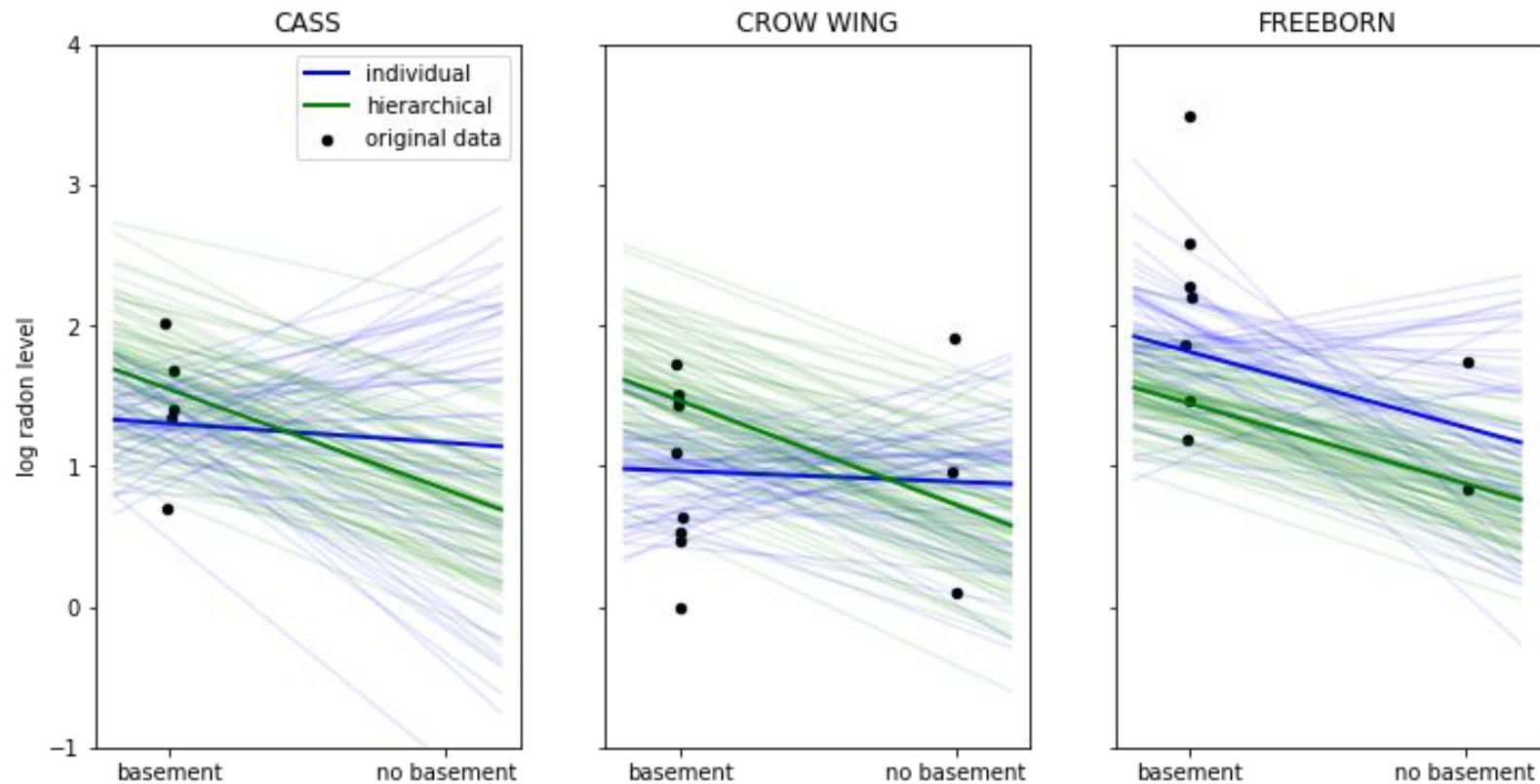
Radon is a toxic gas that seeps into homes from the ground and is the number one cause of lung cancer outside of smoking.

Radon levels were measured across counties in Minnesota with a field denoting which floor the measurement was taken.

Not all counties had records with basement measurements



# Case Study: Radon Prediction



## Making your model hierarchical

---

$$y = \alpha + \beta x$$

Simple Model

$$y = \alpha_j + \beta x$$

Varying intercept

$$y = \alpha_j + \beta_j x$$

Varying Slope - Varying  
Intercept

$$y \sim N(\alpha_j + \beta_j x, \sigma_y^2) \quad \begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N \left( \begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 \\ \sigma_\beta^2 \end{pmatrix} \right)$$

# Probabilistic Models in three steps

---

1. Posit priors and declare likelihood estimator
2. Infer values for latent variables
3. Check your model

# Posit priors and declare likelihood estimator

---

```
lr = pm.Model()

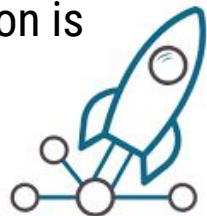
with lr:

    alpha = pm.Normal('alpha', mu=0, sd=10e4, shape=(1))
    betas = pm.Normal('betas', mu=0, sd=10e4, shape=(1, len(X_cols)))
    sigma = pm.HalfNormal('sigma', sd=10e4)

    temp = alpha + T.dot(model_input, betas.T)

    y = pm.Lognormal('y', mu=temp, sd=sigma, observed=model_output)
```

Non-informative priors can be used when no prior information is available, but all possibilities must be represented.



**PYMC3**

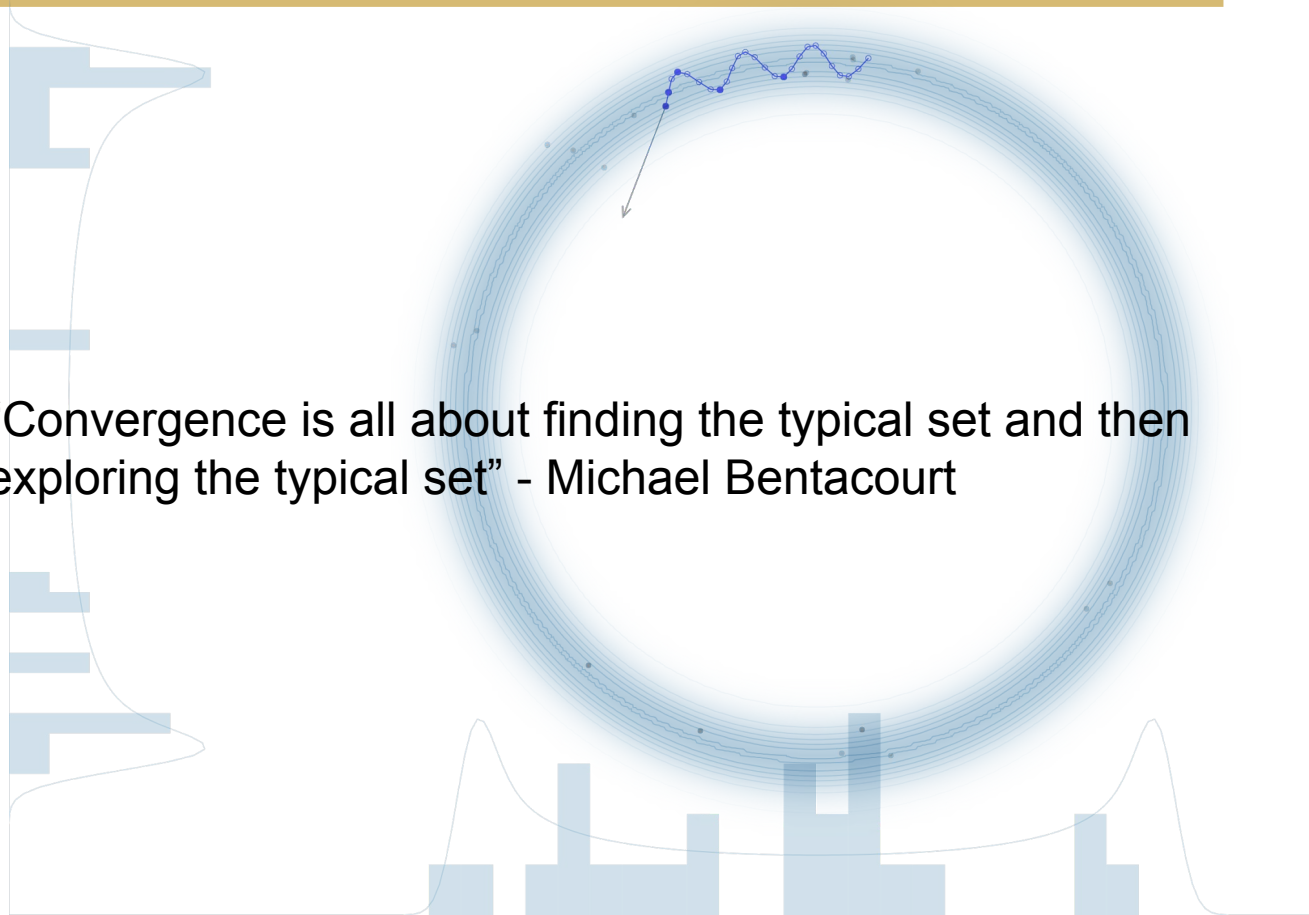
# Infer Values for Latent Variables

---

## No U-Turn Sampler

A Hamiltonian Monte Carlo Markov Chain sampler for efficient sampling

“Convergence is all about finding the typical set and then exploring the typical set” - Michael Bentacourt



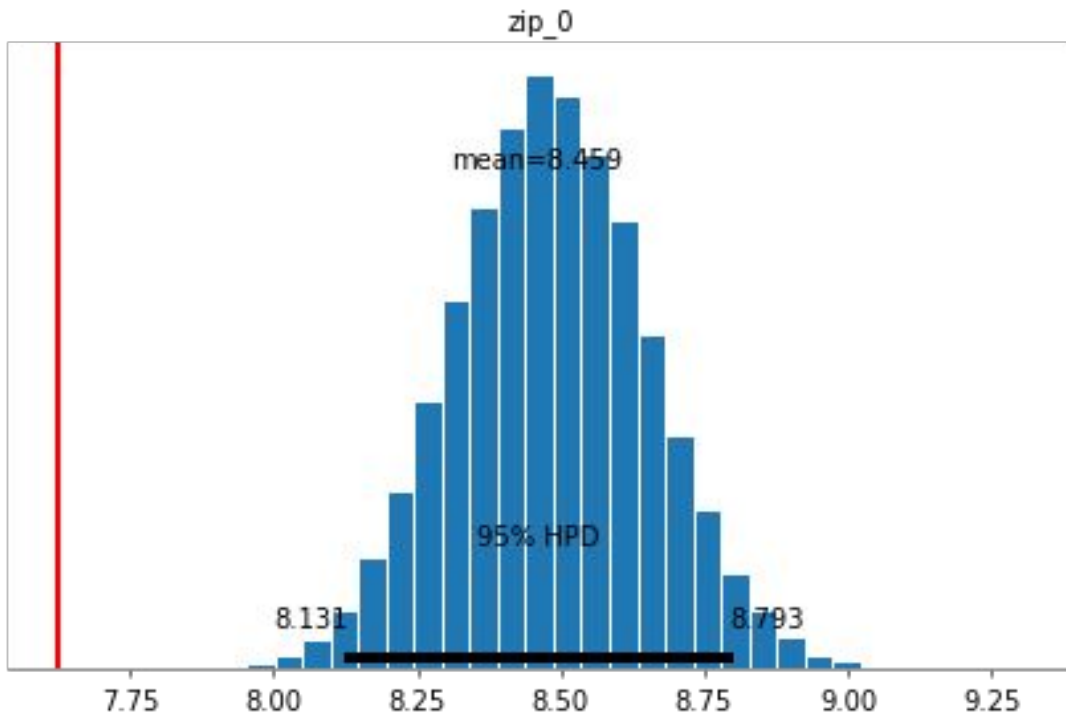
# Check your Model

---

Is your model  
sensitive to  
initialization of new  
priors?

Has your model  
converged?

Could your actual  
data have come  
from your posterior?



# Recap

---

1. A hierarchical model is a probabilistic model which allows for varying slope and intercepts that come from a beta distribution based on group indicators.
2. Probabilistic models can be generated using priors and a likelihood estimator.
3. When there is no prior knowledge a non-informative prior can be used.
4. Model health can be assessed through comparing the posterior distribution to actual data and from the sampling traceplot.



# Putting it all Together

---

PyMC3 and Hierarchical models in practice

## Making your model hierarchical

---

$$y = \alpha + \beta x$$

Simple Model

$$y = \alpha_j + \beta x$$

Varying intercept

$$y = \alpha_j + \beta_j x$$

Varying Slope - Varying  
Intercept

$$y \sim N(\alpha_j + \beta_j x, \sigma_y^2) \quad \begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 \\ \sigma_\beta^2 \end{pmatrix}\right)$$