McGill Computation and Data Systems Initiative

Bayesian Hierarchical Models

Tyrel Stokes
April 11, 2022

tyrel.stokes@mail.mcgill.ca

Overview

1. What are Hierarchical/Bayesian Hierarchical Models
2. Understanding Random Effects vs Fixed Effects
3. Example with NBA Free Throw Data (STAN)
4. Hierarchical Models for Longitudinal Data
5. Quick Example with Covid Case Data (INLA)

Materials (including the code) can be found at the following GitHub repo:
https://github.com/tyrelstokes/BayesianHierarchicalWorkshop

Hierarchical Models

Not Exclusive to Bayesians

Hierarchical Models

Not Exclusive to Bayesians

E.g: Say we have random variables X,Y,Z

Have joint distribution: $p(X = x, Y = y, Z = z; \theta)$

Hierarchical Models

Not Exclusive to Bayesians

E.g: Say we have random variables X,Y,Z

Have joint distribution: $p(X = x, Y = y, Z = z; \theta)$

By Standard probability calculus can decompose into:

$$p(Z = z \mid X = x, Y = y; \theta_1) p(Y = y \mid X = x; \theta_2) p(X = x; \theta_3)$$

Hierarchical Models

Not Exclusive to Bayesians

E.g: Say we have random variables X,Y,Z

Have joint distribution: $p(X = x, Y = y, Z = z; \theta)$

By Standard probability calculus can decompose into:

$$\underbrace{p(Z = z \mid X = x, Y = y; \theta_1)}_{\text{Model 1}}\underbrace{p(Y = y \mid X = x; \theta_2)}_{\text{Model 2}}\underbrace{P(X = x; \theta_3)}_{\text{Model 3}}$$

Bayesian Hierarchical Models

Bayesian Hierarchical Models need not be hierarchical at the level of the likelihood

Recall that unknown parameters are treated as
random variables in the bayesian paradigm

Hierarchical Example

Y_{ij} : Average SAT score of school j in district i

Hierarchical Example

Y_{ij} : Average SAT score of school j in district i

$$Y_{ij} \sim N(\mu_i, \sigma)$$

Likelihood

Hierarchical Example

Y_{ij} : Average SAT score of school j in district i

$$Y_{ij} \sim N(\mu_i, \sigma)$$

Likelihood

$$\mu_i \sim N(\mu, \tau), \forall i$$

(District Level Mean Prior)

Hierarchical Example

Y_{ij} : Average SAT score of school j in district i

$Y_{ij} \sim N(\mu_i, \sigma)$ Likelihood

$\mu_i \sim N(\mu, \tau), \forall i$ (District Level Mean Prior)

$\mu \sim N(1600,500)$ (Overall Mean Hyperprior)

$\tau \sim$ Half Normal(0,200) (Between district stdev Hyperprior)

$\sigma \sim$ Half Normal(0,300) (Between School Variation Prior)

Hierarchical Example

Model

$$Y_{ij} \sim N(\mu_i, \sigma)$$

$$\mu_i \sim N(\mu, \tau), \forall i$$

$$\mu \sim N(1600,500)$$

$$\tau \sim \text{Half Normal}(0,200)$$

$$\sigma \sim \text{Half Normal}(0,300)$$

Hierarchical Example

Model

$$Y_{ij} \sim N(\mu_i, \sigma)$$

$$\mu_i \sim N(\mu, \tau), \forall i$$

$$\mu \sim N(1600,500)$$

$$\tau \sim \text{Half Normal}(0,200)$$

$$\sigma \sim \text{Half Normal}(0,300)$$

$$\overbrace{p(\mu_i, \mu, \sigma, \tau \mid Y)}^{\text{Posterior}} \propto \prod_i \prod_j \overbrace{p(Y_{ij} \mid \mu_i, \mu, \sigma, \tau)}^{\text{Likelihood}} \overbrace{\pi(\mu_i, \mu, \sigma, \tau)}^{\text{Prior}}$$

Hierarchical Example

Model

$$Y_{ij} \sim N(\mu_i, \sigma)$$

$$\mu_i \sim N(\mu, \tau), \forall i$$

$$\mu \sim N(1600,500)$$

$$\tau \sim \text{Half Normal}(0,200)$$

$$\sigma \sim \text{Half Normal}(0,300)$$

$$\overbrace{p(\mu_i, \mu, \sigma, \tau \mid Y)}^{\text{Posterior}} \propto \prod_i \prod_j \overbrace{p(Y_{ij} \mid \mu_i, \mu, \sigma, \tau)}^{\text{Likelihood}} \overbrace{\pi(\mu_i, \mu, \sigma, \tau)}^{\text{Prior}}$$

$$= \prod_i \prod_j p(Y_{ij} \mid \mu_i, \mu, \sigma, \tau) \boxed{\pi(\mu_i \mid \mu, \sigma, \tau)\pi(\mu, \sigma, \tau)}$$

Hierarchical Example

Model

$$Y_{ij} \sim N(\mu_i, \sigma)$$

$$\mu_i \sim N(\mu, \tau), \forall i$$

$$\mu \sim N(1600,500)$$

$$\tau \sim \text{Half Normal}(0,200)$$

$$\sigma \sim \text{Half Normal}(0,300)$$

$$\overbrace{p(\mu_i, \mu, \sigma, \tau \mid Y)}^{\text{Posterior}} \propto \prod_i \prod_j \overbrace{p(Y_{ij} \mid \mu_i, \mu, \sigma, \tau)}^{\text{Likelihood}} \overbrace{\pi(\mu_i, \mu, \sigma, \tau)}^{\text{Prior}}$$

$$= \prod_i \prod_j p(Y_{ij} \mid \mu_i, \mu, \sigma, \tau)\pi(\mu_i \mid \mu, \sigma, \tau)\pi(\mu, \sigma, \tau)$$

$$= \prod_i \prod_j p(Y_{ij} \mid \mu_i, \sigma)\pi(\mu_i \mid \mu, \tau)\pi(\mu)\pi(\sigma)\pi(\tau)$$

Random Effect Models

The previous model is an example of a "random effects" model

This kind of model goes by many names in different statistics cultures

- Mixed Effects , nested model, multilevel model, etc
    - See this blog post for more translations

$$\mu_i \sim N(\mu, \tau), \forall i$$

We think of some of the parameters as being "drawn" from a larger population

Random Effect Models

$$\mu_i \sim N(\mu, \tau), \forall i$$

A few perspectives on the Random Effects Model

- A compromise between a “pooled” and fixed effect model
- Exchangeability
- Unobserved heterogeneity (variance or uncertainty perspective)

Compromise between Pooled and Fixed

$$\mu_i \sim N(\mu, \tau), \forall i$$

Pooled
All Groups share overall mean

Fixed Effects
Each Group has separate mean

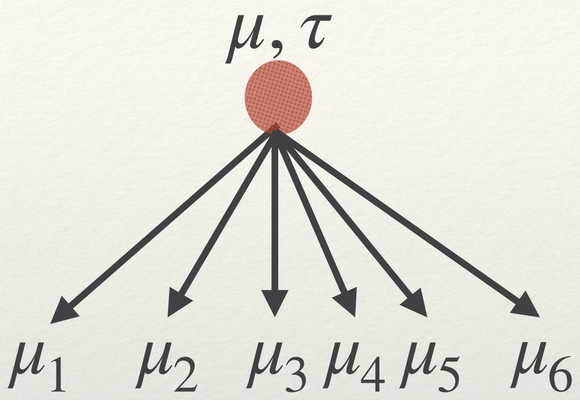
$\tau \to 0$

τ

$\tau \to \infty$

Between group variance

Exchangeability



Random Variables are **exchangeable** when the joint probability is **invariant to (finite) permutations**

Exchangeability Example

Suppose we have random variables X_1, X_2, X_3

Where X_i is the free throw percentage of player i

Suppose we **don't know which player is which**, just that they represent different players

Exchangeability Example

Suppose we have random variables X_1, X_2, X_3

Where X_i is the free throw percentage of player i

Suppose we **don't know which player is which**, just that they represent different players

---

Would you model the following differently?

Exchangeability Example

Suppose we have random variables X_1, X_2, X_3

Where X_i is the free throw percentage of player i

Suppose we **don't know which player is which**, just that they represent different players

---

Would you model the following differently?

1. $\boldsymbol{f(X_1 = x_1, X_2 = x_2, X_3 = x_3)}$

Exchangeability Example

Suppose we have random variables X_1, X_2, X_3

Where X_i is the free throw percentage of player i

Suppose we **don't know which player is which**, just that they represent different players

---

Would you model the following differently?

1. $f(X_1 = x_1, X_2 = x_2, X_3 = x_3)$
2. $f(X_1 = x_2, X_2 = x_3, X_3 = x_1)$

Exchangeability Example

Stephen Curry
Career Avg FT pct: **90.8%**

Exchangeability Example

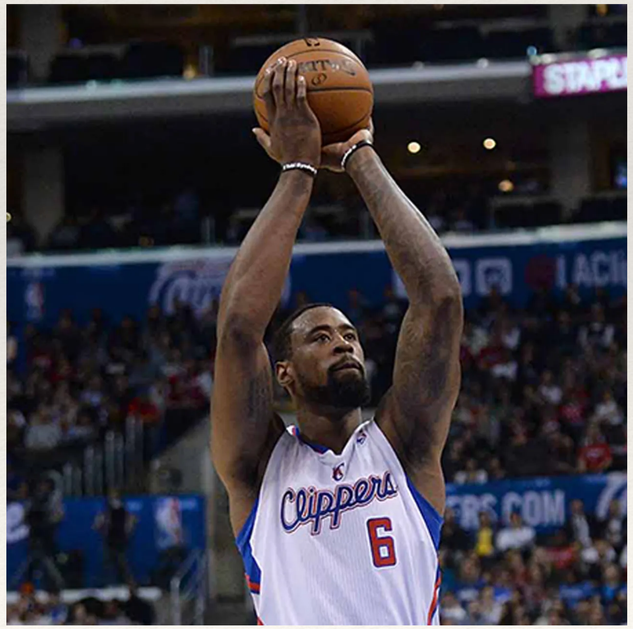
Stephen Curry
Career Avg FT pct: **90.8%**

Lebron James
Career Avg FT pct: **73.4%**

Exchangeability Example

Stephen Curry
Career Avg FT pct: **90.8%**

Lebron James
Career Avg FT pct: **73.4%**

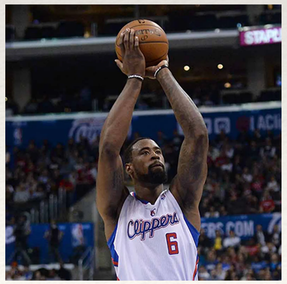


Deandre Jordan
Career Avg FT pct: **47.6%**

Exchangeability Example

Stephen Curry
Career Avg FT pct: **90.8%**

Lebron James
Career Avg FT pct: **73.4%**



Deandre Jordan
Career Avg FT pct: **47.6%**

X_{Curry}: Stephen Curry's Free Throw percentage in some season

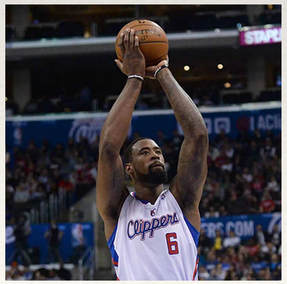
X_{James}: Lebron James' Free Throw percentage in some season

X_{Curry}: Deandre Jordan's Free Throw percentage in some season

Exchangeability Example

Stephen Curry
Career Avg FT pct: **90.8%**

Lebron James
Career Avg FT pct: **73.4%**



Deandre Jordan
Career Avg FT pct: **47.6%**

X_{Curry}: Stephen Curry's Free Throw percentage in some season

X_{James}: Lebron James' Free Throw percentage in some season

X_{Curry}: Deandre Jordan's Free Throw percentage in some season

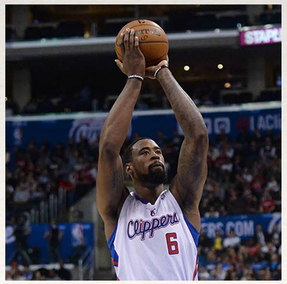
Now would you say:

$$f(X_{Curry} = 0.9, \quad X_{James} = 0.7, \quad X_{Jordan} = 0.5)$$

Exchangeability Example

Stephen Curry
Career Avg FT pct: **90.8%**

Lebron James
Career Avg FT pct: **73.4%**



Deandre Jordan
Career Avg FT pct: **47.6%**

X_{Curry}: Stephen Curry's Free Throw percentage in some season

X_{James}: Lebron James' Free Throw percentage in some season

X_{Curry}: Deandre Jordan's Free Throw percentage in some season

Now would you say:

$$f(X_{Curry} = 0.9, \quad X_{James} = 0.7, \quad X_{Jordan} = 0.5)$$

$$=$$

$$f(X_{Curry} = 0.5, \quad X_{James} = 0.9, \quad X_{Jordan} = 0.7)$$

Exchangeability and De Finetti

If random variables $(X_1, \ldots, X_n)$ are exchangeable

Then there exists a parametric model $p(x \mid \theta)$ such that:

$$p(x_1, \ldots, x_n) = \int_{\theta \in \Theta} \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) d(\theta)$$

Exchangeability and De Finetti

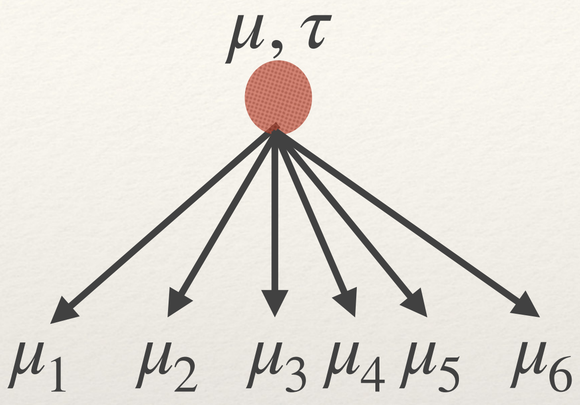
If random variables $(X_1, \ldots, X_n)$ are exchangeable

Then there exists a parametric model $p(x \mid \theta)$ such that:

$$p(x_1, \ldots, x_n) = \int_{\theta \in \Theta} \underbrace{\prod_{i=1}^{n} p(x_i \mid \theta)}_{\text{Looks a lot like a likelihood}} \underbrace{p(\theta) d(\theta)}_{\text{Looks a lot like a Prior}}$$

Exchangeability

$$p(x_1, \ldots, x_n) = \int_{\theta \in \Theta} \prod_{i=1}^{n} p(x_i \mid \theta) p(\theta) d(\theta)$$



Recalling the random effects example

$$p(\mu_1, \ldots, \mu_6, \mu, \tau, \sigma) = \prod_{i=1}^{6} p(\mu_i \mid \mu, \tau) p(\mu) p(\tau) p(\sigma)$$

Random Effects Example (Binomial Likelihood)

Y_{it} = Number of made free throws by player i in season t

N_{it} = Number of attempted free throws by player i in season t

Random Effects Example (Binomial Likelihood)

Y_{it} = Number of made free throws by player i in season t

N_{it} = Number of attempted free throws by player i in season t

$$Y_{it} \sim Binomial(\theta_i; N_{it})$$

Likelihood

Random Effects Example (Binomial Likelihood)

Y_{it} = Number of made free throws by player i in season t

N_{it} = Number of attempted free throws by player i in season t

$$Y_{it} \sim Binomial(\theta_i; N_{it})$$

Likelihood

$$\theta_i = \frac{1}{1 + \exp(-\mu_i)}$$

Player Mean

Random Effects Example (Binomial Likelihood)

Y_{it} = Number of made free throws by player i in season t

N_{it} = Number of attempted free throws by player i in season t

$$Y_{it} \sim Binomial(\theta_i; N_{it})$$

Likelihood

$$\theta_i = \frac{1}{1 + \exp(-\mu_i)}$$

Player Mean

$$\mu_i \sim N(\mu, \tau)$$

Random Effect Prior

Random Effects Example (Binomial Likelihood)

Y_{it} = Number of made free throws by player i in season t

N_{it} = Number of attempted free throws by player i in season t

$$Y_{it} \sim Binomial(\theta_i; N_{it})$$ Likelihood

$$\theta_i = \frac{1}{1 + \exp(-\mu_i)}$$ Player Mean

$$\mu_i \sim N(\mu, \tau)$$ Random Effect Prior

$$\mu \sim N(0,5)$$ Overall Mean Hyperprior

$$\tau \sim N(0,3)$$ Between Player Variance Hyperprior

Data Snippet

player	season	attempts	makes	made_pct
A.J. Price	2014 - 2015	17	11	0.65
Aaron Brooks	2014 - 2015	181	149	0.82
Aaron Gordon	2014 - 2015	61	44	0.72
Adreian Payne	2014 - 2015	46	30	0.65
Al Horford	2014 - 2015	161	122	0.76
Al Jefferson	2014 - 2015	165	108	0.65
Al-Farouq Aminu	2014 - 2015	137	99	0.72
Alan Anderson	2014 - 2015	110	88	0.80
Alec Burks	2014 - 2015	129	106	0.82
Alex Kirk	2014 - 2015	2	2	1.00
Alex Len	2014 - 2015	104	73	0.70
Alexey Shved	2014 - 2015	141	114	0.81
Alexis Ajinca	2014 - 2015	99	81	0.82
Allen Crabbe	2014 - 2015	12	9	0.75

Goal: Use 2014-15 Season to make 2015-16 predictions

475 Players took free throws in 2015-16

465 Players took free throws in 2014-15

389 Players took free throws in both seasons

That leaves **86** players in prediction set (2015-16) without data in training set

STAN

Probabilistic Language for Bayesian Inference

Uses **MCMC** to draw samples to estimate posterior

Specifically Hamiltonian Monte Carlo

Philosophy is to **prioritize flexibility** in modelling over optimization for specific cases

Many ways to use it - In R there are two packages **Rstan** and **Cmdstanr**

More user friendly versions - **brms** and **rstanarm**

Implementations in other languages/platforms - shell, python, Matlab, Julia, stata, etc.

5 Minute Break

Stan Code

```
data {
// All of the necessary data goes here
}
parameters {
// Any parameter which we directly put a prior on goes here
}
transformed parameters{
  // Any transformation of the above parameters which we'd like to store
  // goes here
}
model {
// The likelihood and prior goes here
// and any useful transformations that we do not want to store
}
generated quantities{
  // This is where we calculate quantities of interest, such
  // as predictions

}
```

Stan Code - Data Block

```
data {
  int<lower=0> n_train;// The number of data points in training data (2014-15)
  int<lower=0> n_test;// The number of data points in prediction set (2015-16)

  int y[n_train] ;// The number of made free throws in training data (2014-15)
  int y_test[n_test];// The number of free throws made in test (2015-16)

  int<lower=0> Attempts[n_train]; //The number of attempted free throws in training (2014-15)
  int<lower=0> Attempts_test[n_test];// number of attempted in test (2015-16)

  int<lower=0> player_training_int[n_train];
  int<lower =0> player_test_int[n_test];


  int in_training[n_test];// Binary variable 1 if player is in training, 0 otherwise

  int np; // Number of players in training data (2014-15)

  real tau_std_prior;// The variance for between variance prior
  real mu_std_prior;// The variance for the group mean prior
}
```

51:34 (Top Level) Stan

Stan Code - Parameter/Transformed Block

```
parameters {
  real mu;// group mean
  vector[np] mu_0i;// demeaned individual player mean, mu_i = mu + mu_0i*tau
  real<lower=0> tau;// Between player mean
}
transformed parameters{

 vector[np] theta;

 for(i in 1:np){
 theta[i] = 1.0/(1.0+exp(-(mu + mu_0i[i]*tau)));
 }
}
```

Stan Code - Parameter/Transformed Block

```
parameters {
  real mu;// group mean
  vector[np] mu_0i;// demeaned individual player mean, mu_i = mu + mu_0i*tau
  real<lower=0> tau;// Between player mean
}
transformed parameters{

 vector[np] theta;

 for(i in 1:np){
 theta[i] = 1.0/(1.0+exp(-(mu + mu_0i[i]*tau)));
 }
}
```

NB: For computational reasons we reparametrized the model

Instead of $\mu_i \sim N(\mu, \tau)$, we write $\mu_i = \mu + \mu_i^0 \times \tau$

Where $\mu_i^0 \sim N(0,1)$

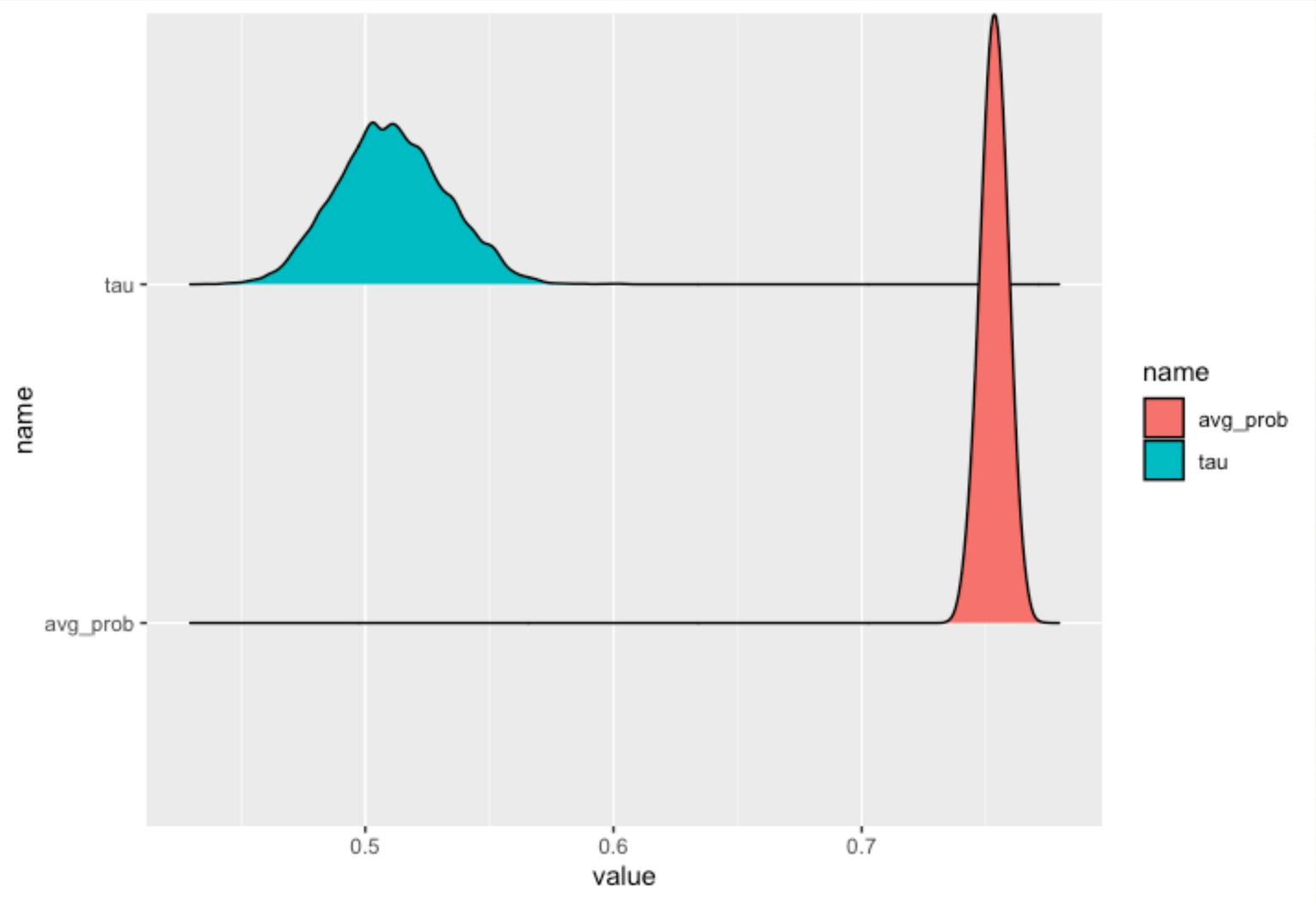
Stan Code - Model Block

```
model {
  y ~ binomial(Attempts,theta);// likelihood

  mu_0i ~ normal(0.0,1.0);//standardized player level prior

  mu ~ normal(0.0,mu_std_prior); // group mean prior
  tau ~ normal(0.0,tau_std_prior);// between player std prior
}
```

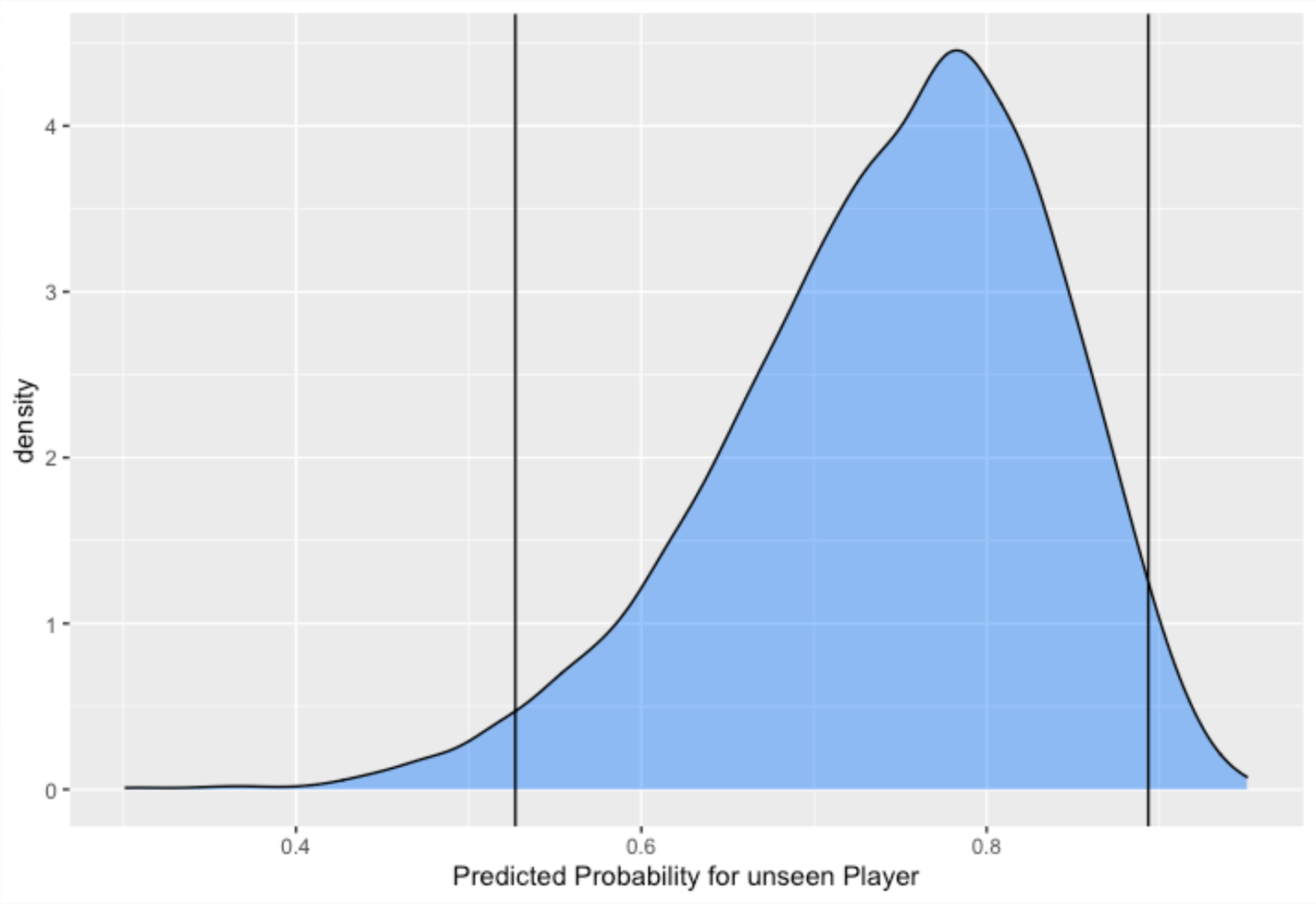
Stan Code - Generated Quantities Block

```
generated quantities{

  vector[n_test] predicted_makes;
  vector[n_test] predicted_probability;

  real avg_prob = 1/(1+exp(-mu));
  real mu_draw;

  for(i in 1:n_test){
    if(in_training[i]==1){
      predicted_makes[i] = binomial_rng(Attempts_test[i],theta[player_test_int[i]]);
      predicted_probability[i] = theta[player_test_int[i]];

    }else{
      mu_draw = normal_rng(mu,tau);
      predicted_probability[i] = 1.0/(1.0+exp(-mu_draw));
      predicted_makes[i] = binomial_rng(Attempts_test[i],predicted_probability[i]);
    }

  }
}
```

Hyper-Prior Results

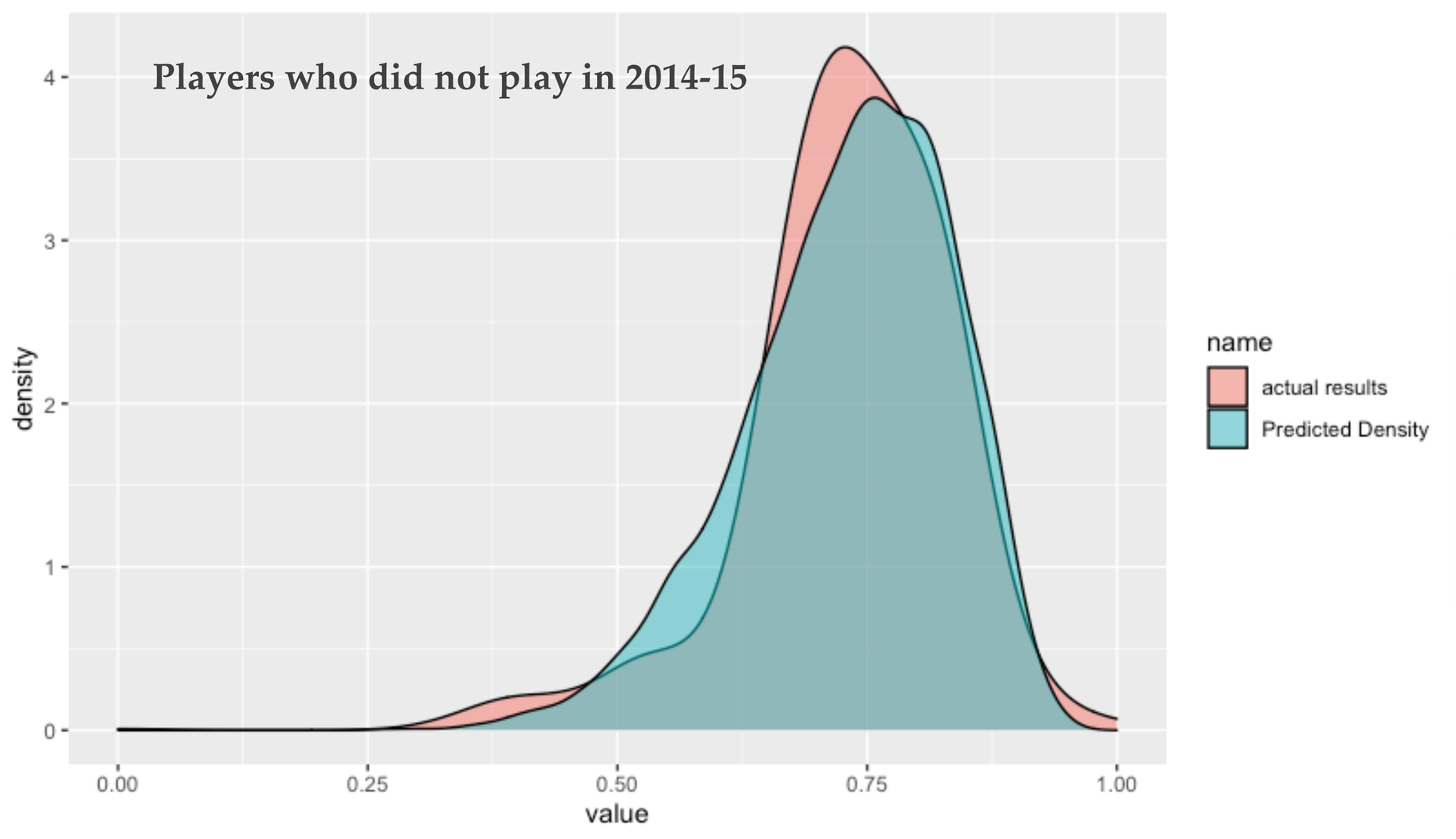


Prediction for unseen player

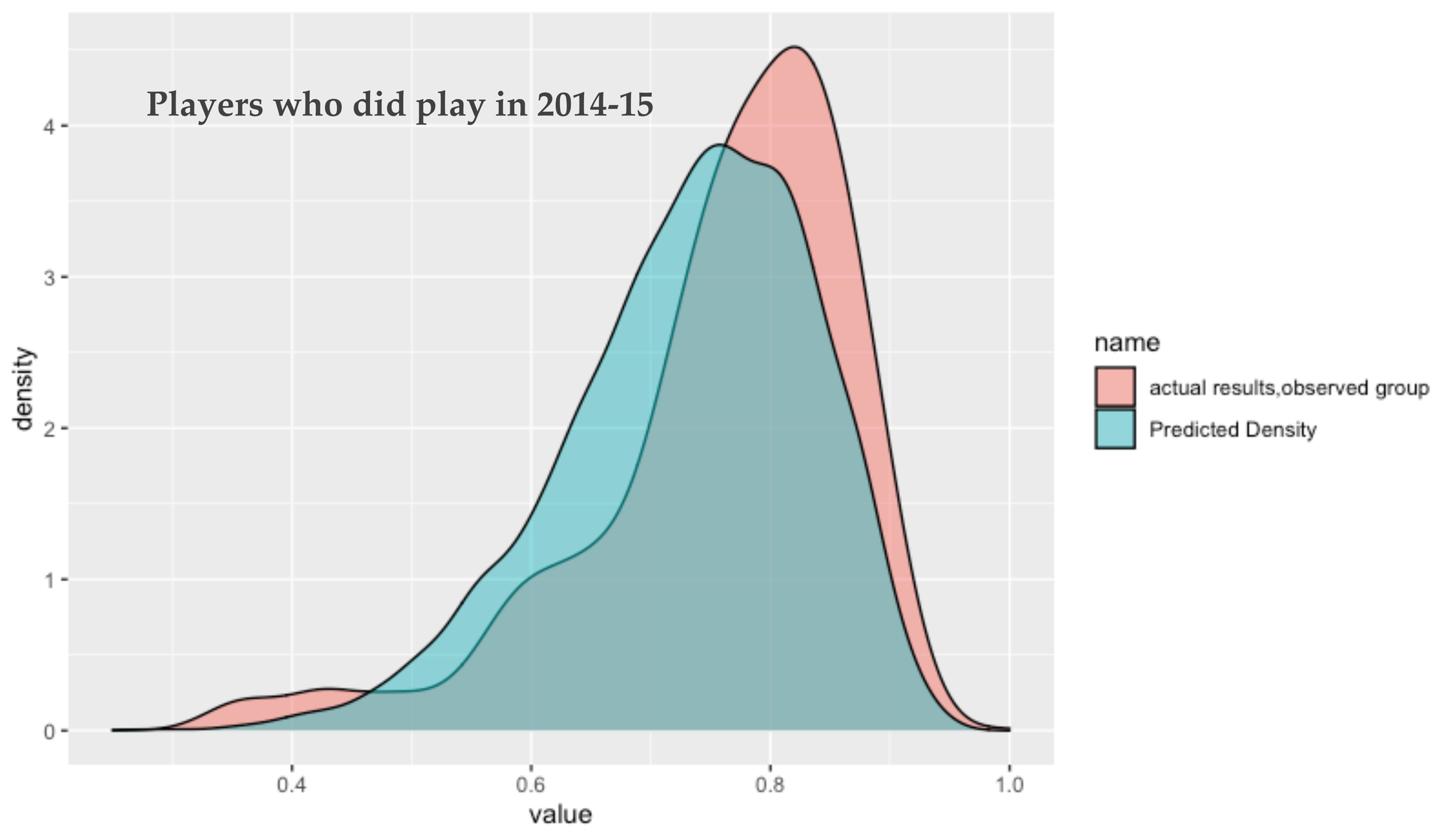


Where $(q_{2.5}, q_{97.5}) = (0.527, 0.894)$

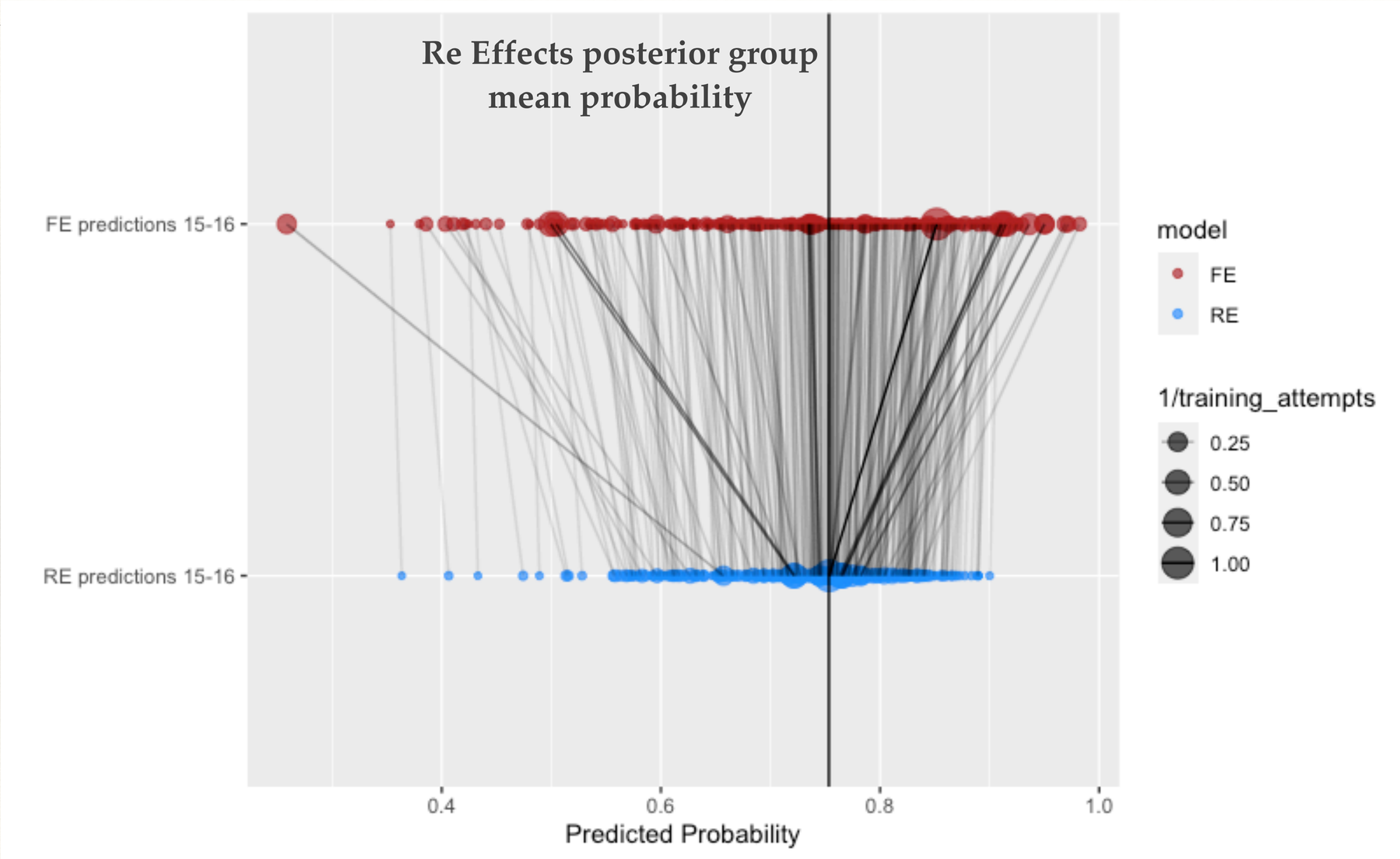
Compare predicted and actual make Probability



Compare predicted and actual make Probability



Random Effects as Regularization to the group mean



Model Comparisons

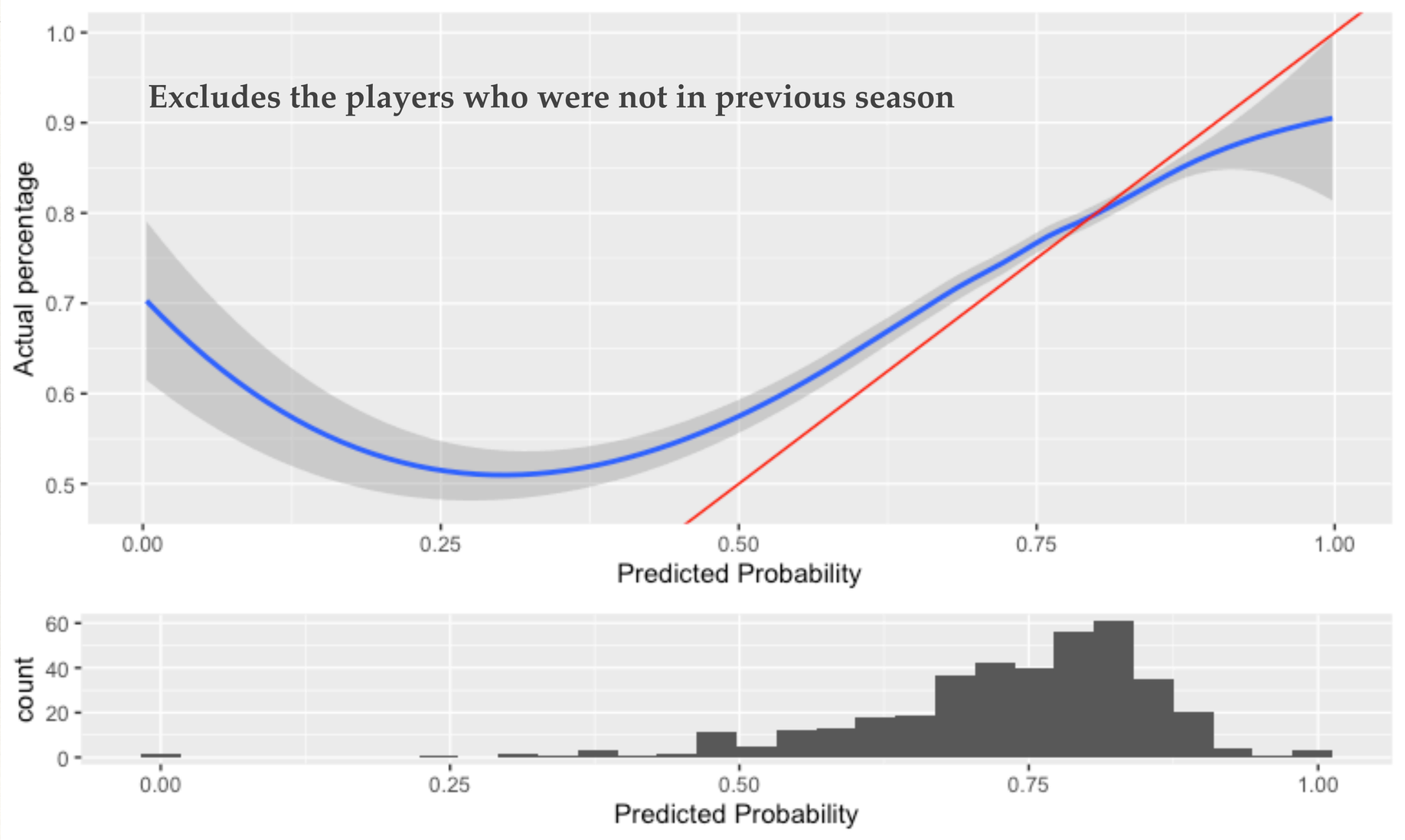
An approximate Leave-One-Out Cross Validation Metric on only 2014-15 data favours the FE model

---

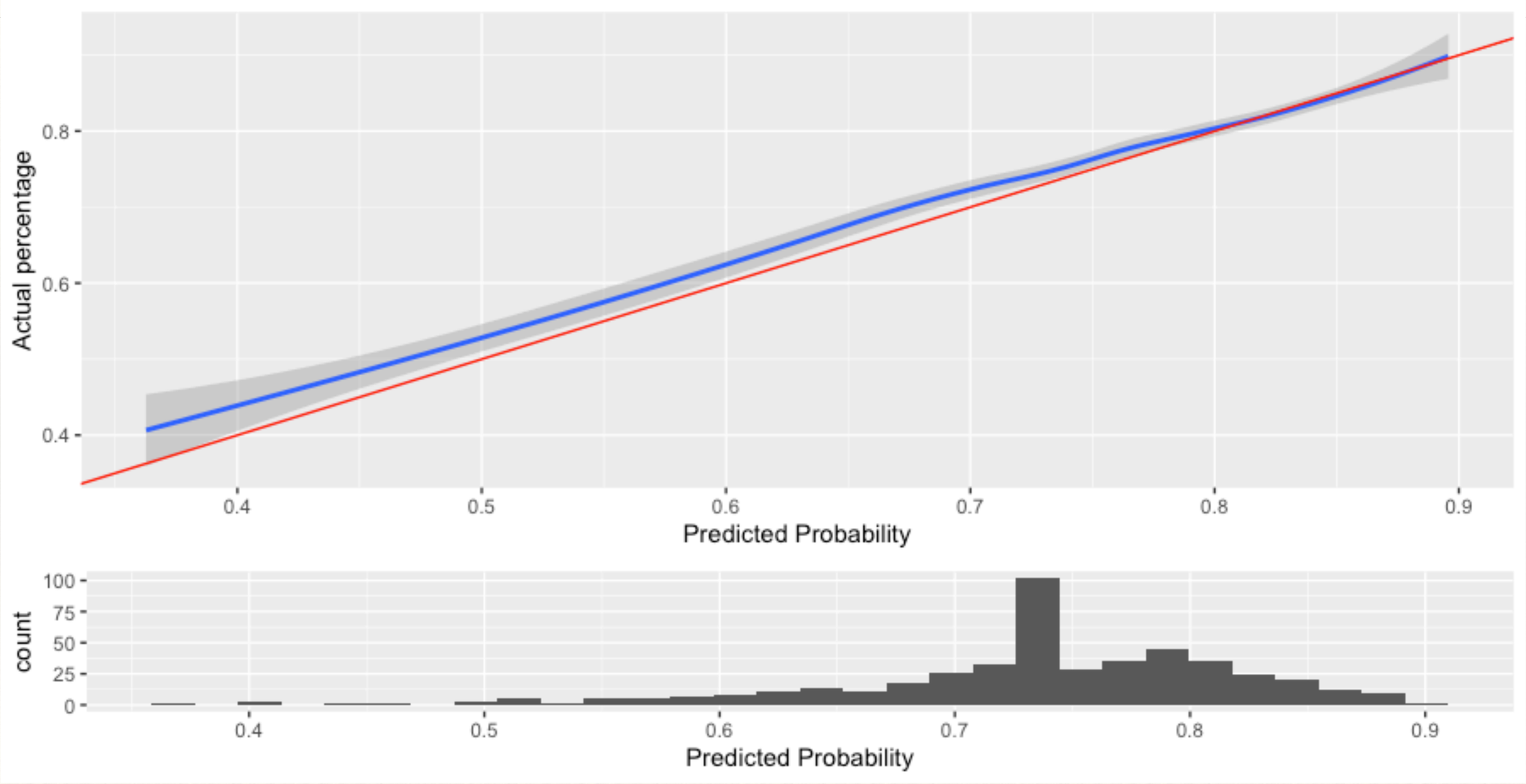
A few checks on out of sample

1. Calibration Plot
2. RMSE

Calibration FE Model on 2015-16 Data



Calibration RE Model on 2015-16 Data



RMSE Comparison

Calculate the Posterior Predicted Mean number of Free Throw makes in 2015-16 given 2014-15 data + number of attempts for each player

RMSE Comparison

Calculate the Posterior Predicted Mean number of Free Throw makes in 2015-16 given 2014-15 data + number of attempts for each player

$$E[Y_{i,16} \mid Y^n]$$

Implicitly average over all of the parameters

RMSE Comparison

Calculate the Posterior Predicted Mean number of Free Throw makes in 2015-16 given 2014-15 data + number of attempts for each player

$$E[Y_{i,16} \mid Y^n]$$

Implicitly average over all of the parameters

Then calculate the RMSE using this predicted value

RMSE Comparison

Calculate the Posterior Predicted Mean number of Free Throw makes in 2015-16 given 2014-15 data + number of attempts for each player

$$E[Y_{i,16} \mid Y^n]$$

Implicitly average over all of the parameters

Then calculate the RMSE using this predicted value

FE RMSE: 13.45
RE RMSE (Only players in both years): **10.63**
RE RMSE (all data): 10.14

Sensitivity to Priors

Models are typically less sensitive to hyper-priors than priors

When exchangeability is reasonable, this means that RE models can offer a form of Protection against poorly calibrated priors compared to the FE model

Prior Experiment

Fixed Effects Model

Random Effects Model

Want to compare two regimes

1. Vague priors
2. Priors which are “close” to the pooled model

“Vague Priors”

Fixed Effects Model

$$\mu_i \sim N(0,100)$$

Random Effects Model

$$\mu \sim N(0,100)$$

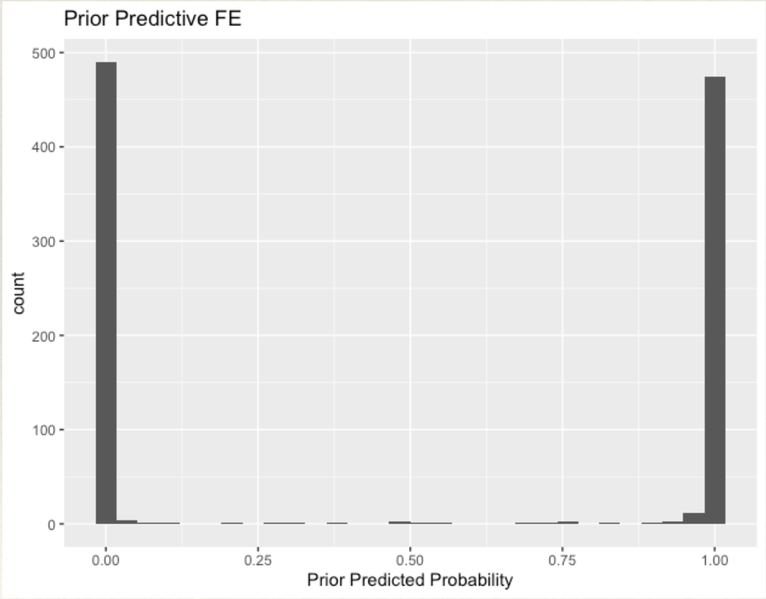
$$\tau \sim \text{Half Normal}(0,100)$$

$$\mu_i \sim N(\mu, \tau)$$

"Vague Priors"

Fixed Effects Model

$$\mu_i \sim N(0,100)$$

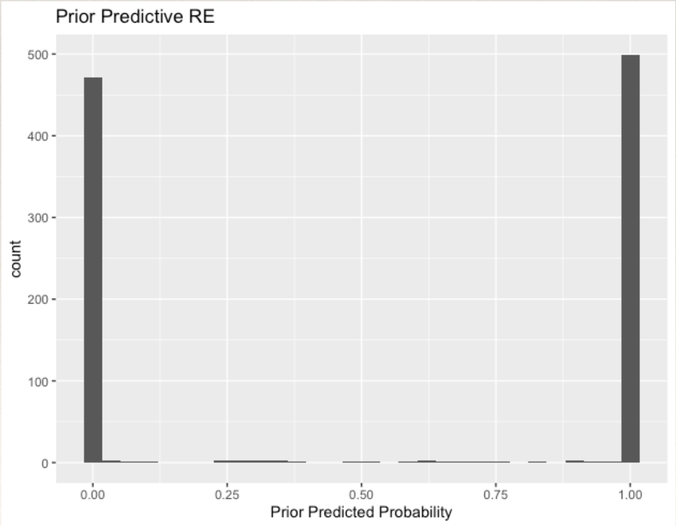


Random Effects Model

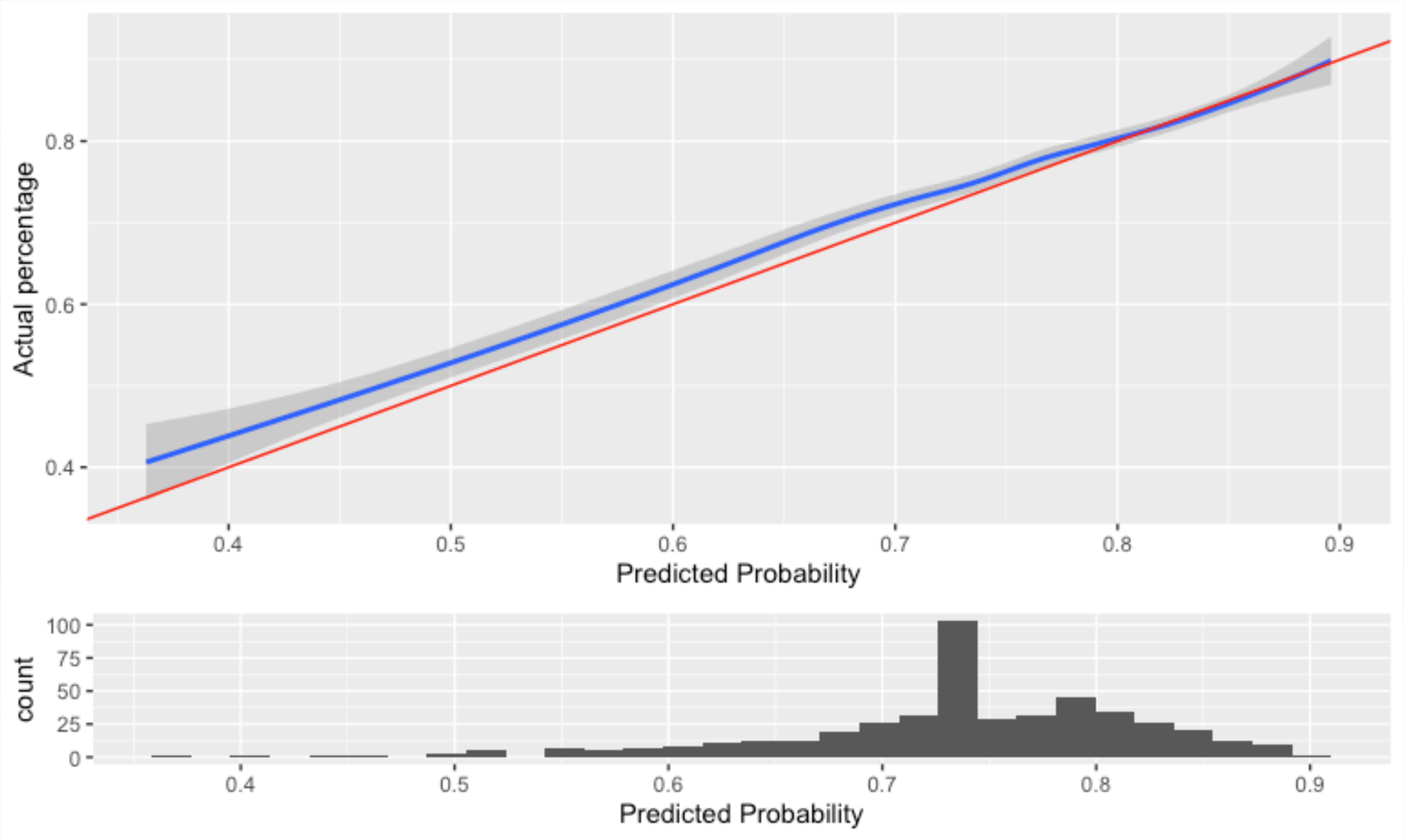
$$\mu \sim N(0,100)$$

$$\tau \sim \text{Half Normal}(0,100)$$

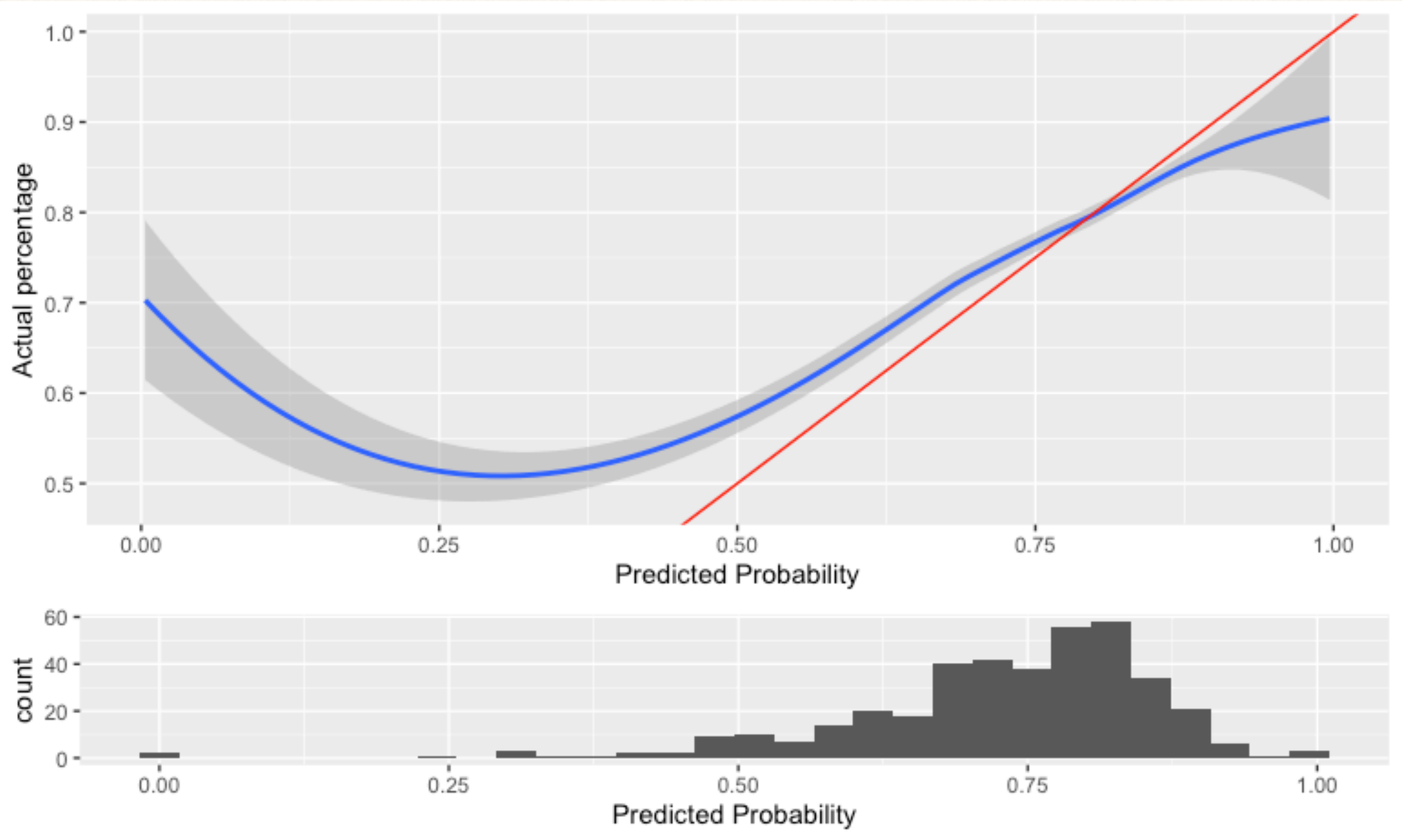
$$\mu_i \sim N(\mu, \tau)$$



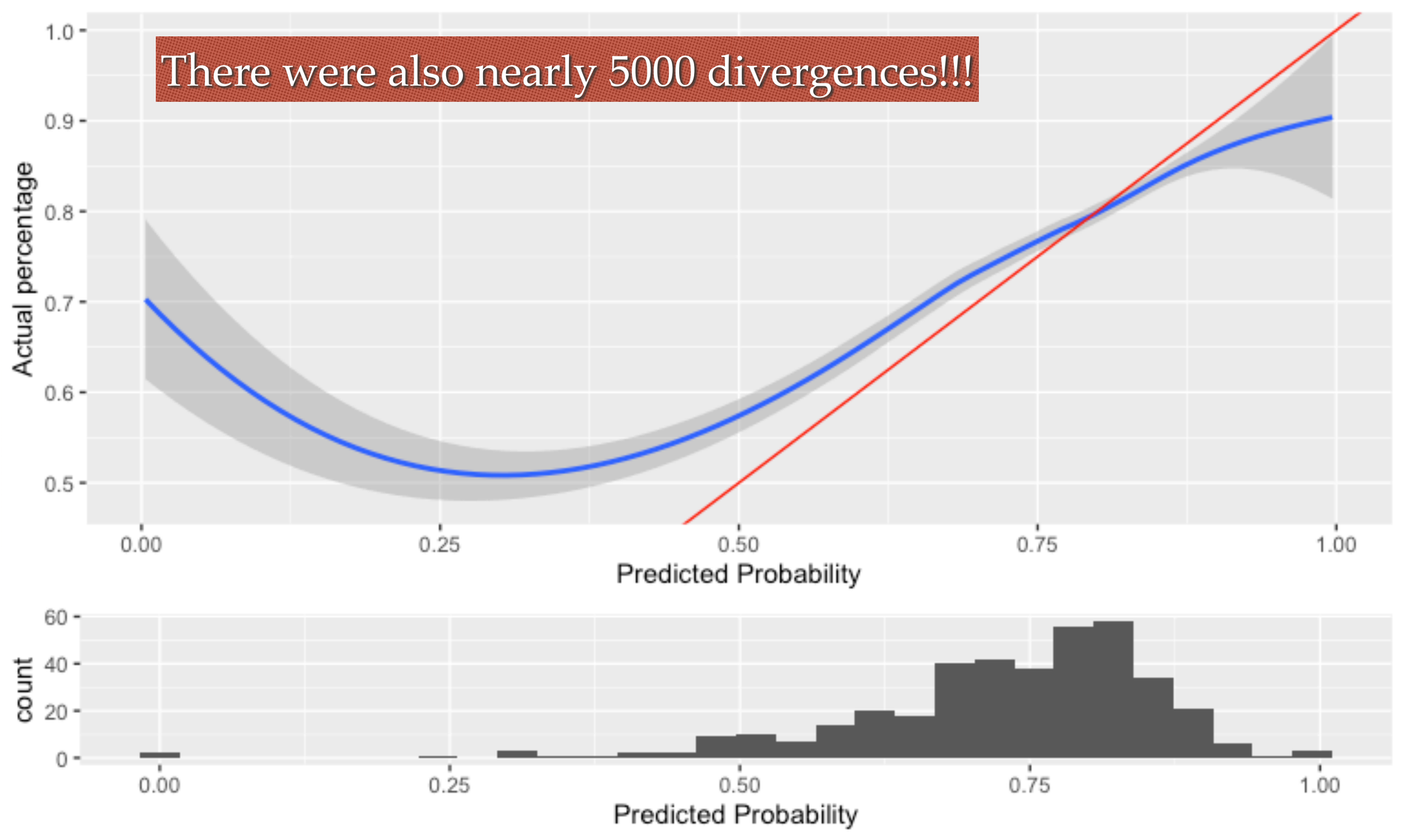
Calibration for Random Effects (Vague Prior)



Calibration for Fixed Effects (Vague Prior)



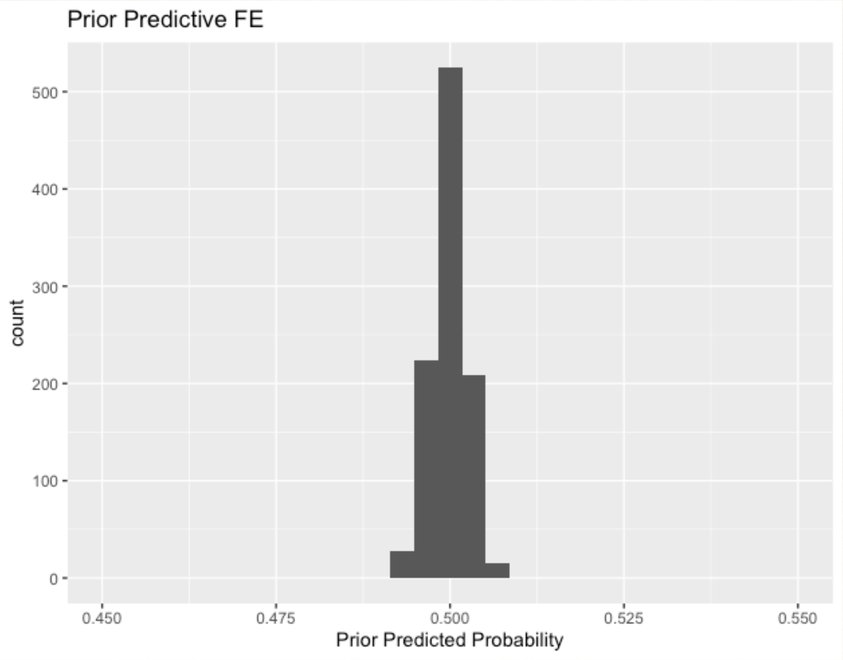
Calibration for Fixed Effects (Vague Prior)



Nearly Pooled Priors

Fixed Effects Model

$$\mu_i \sim N(0,0.01)$$

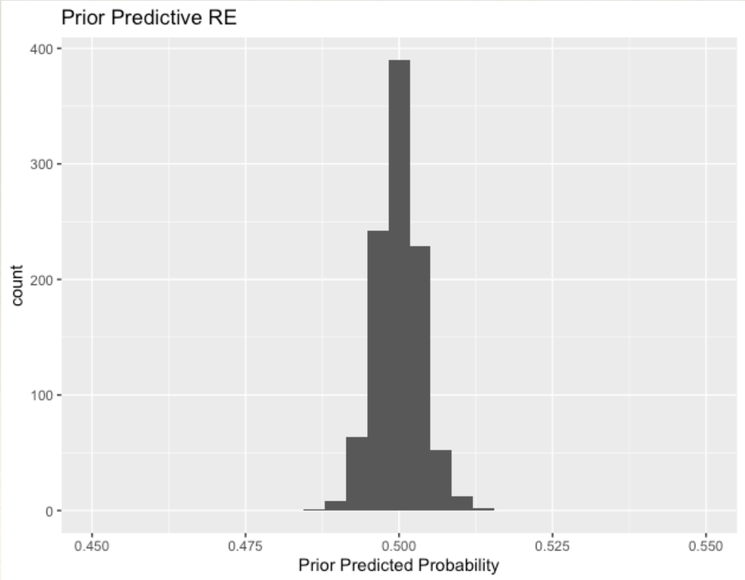


Random Effects Model

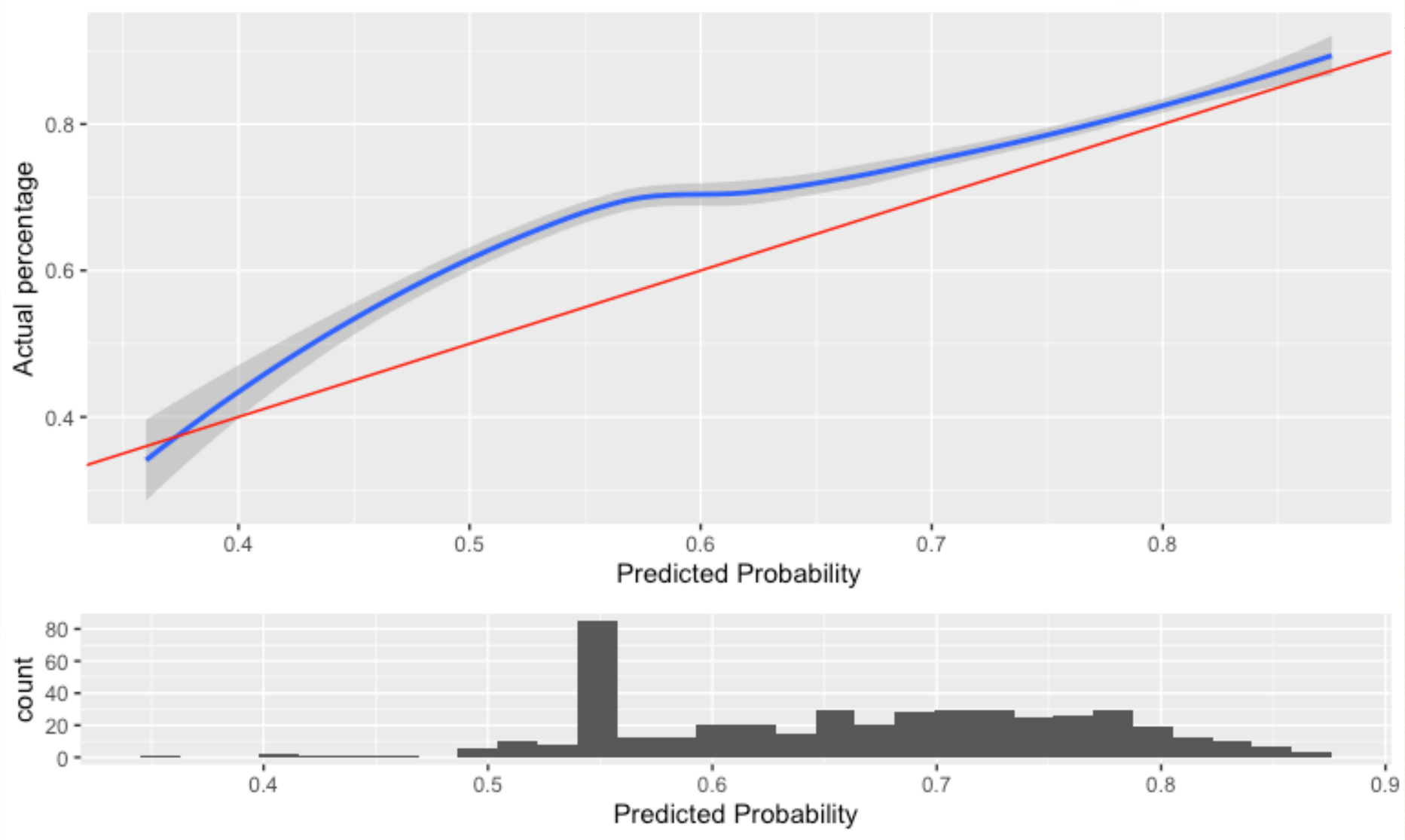
$$\mu \sim N(0,0.01)$$

$$\tau \sim \text{Half Normal}(0,0.01)$$

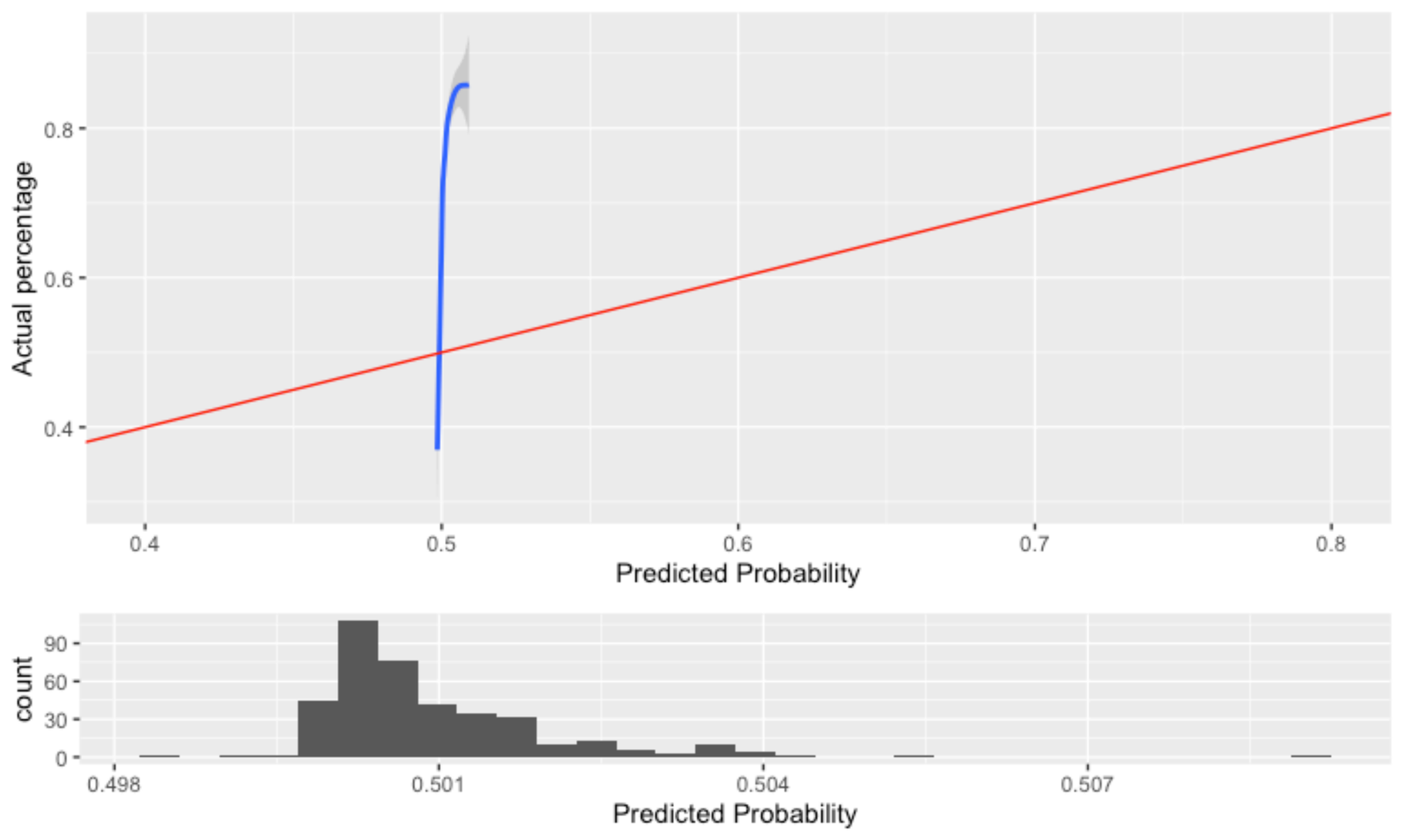
$$\mu_i \sim N(\mu, \tau)$$



Calibration for Random Effects (Tight Prior)



Calibration for Fixed Effects (Tight Prior)



Penalized Complexity Priors

$$\mu_i \sim N(\mu, \tau), \forall i$$

Pooled
All Groups share overall mean

Fixed Effects
Each Group has separate mean

$\tau \to 0$

τ

$\tau \to \infty$

Between group variance

Idea: Penalize the "Distance" from the simpler base model (Occam's Razor)

Hierarchical Models for longitudinal Data

Longitudinal data (panel data) has both **groups** (perhaps exchangeable) AND **time** labels for each group

Within each groups, the **time labels may not be exchangeable**

We can use hierarchical models and ideas from time-series analysis to place priors which allow **parameters to vary over time**

Free Throws Revisited

Suppose we have all the free throws from 2007-2016

Have player-season level data for all players that took at least one FT

Free Throws Revisited

Suppose we have all the free throws from 2007-2016

Have player-season level data for all players that took at least one FT

Could imagine the following model

Free Throws Revisited

Suppose we have all the free throws from 2007-2016

Have player-season level data for all players that took at least one FT

Could imagine the following model

Let $\boldsymbol{\mu_{i0}}$ be the latent FT ability for player i right **before they appear in the data**

Free Throws Revisited

Suppose we have all the free throws from 2007-2016

Have player-season level data for all players that took at least one FT

Could imagine the following model

Let $\boldsymbol{\mu_{i0}}$ be the latent FT ability for player i right **before they appear in the data**

We might suppose that the ability right before appearing is **exchangeable**

Free Throws Revisited

$\boldsymbol{\mu_{i0}}$'s being exchangeable could motivate a random effects prior at time 0

i.e

$$\mu_{i0} \sim N(\mu_0, \tau), \forall i$$

But harder to argue that within each player, the latent ability is exchangeable

Time labels can be meaningful

We might expect players to get better and worse slowly over time for example

Free Throws Revisited

Within each player we could use a time series prior to model their progression

E.g: Random Walk prior

$$\mu_{it} \sim N(\mu_{i,t-1}, \sigma), t > 0$$

On average, we expect roughly the same ability as last year

How much we expect that ability to change is determined by σ

$\sigma \to 0$: Ability is constant over time

$\sigma \to \infty$: Ability likely to change drastically

Free Throws Revisited

We can really use any process that we want to model even these latent processes!

AR(p) processes are common and useful

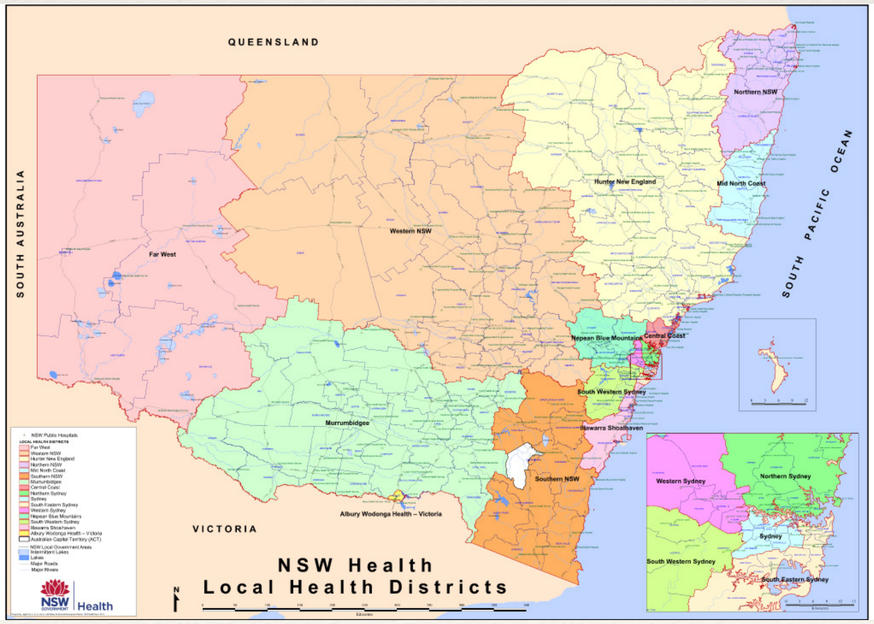
$$\mu_{it} = \mu_{i,t-1}\phi_1 + \ldots + \mu_{i,t-p}\phi_p + \epsilon$$

$$\epsilon \sim N(0,\sigma_t)$$

Models direct dependence with the last p time points

Longitudinal Example

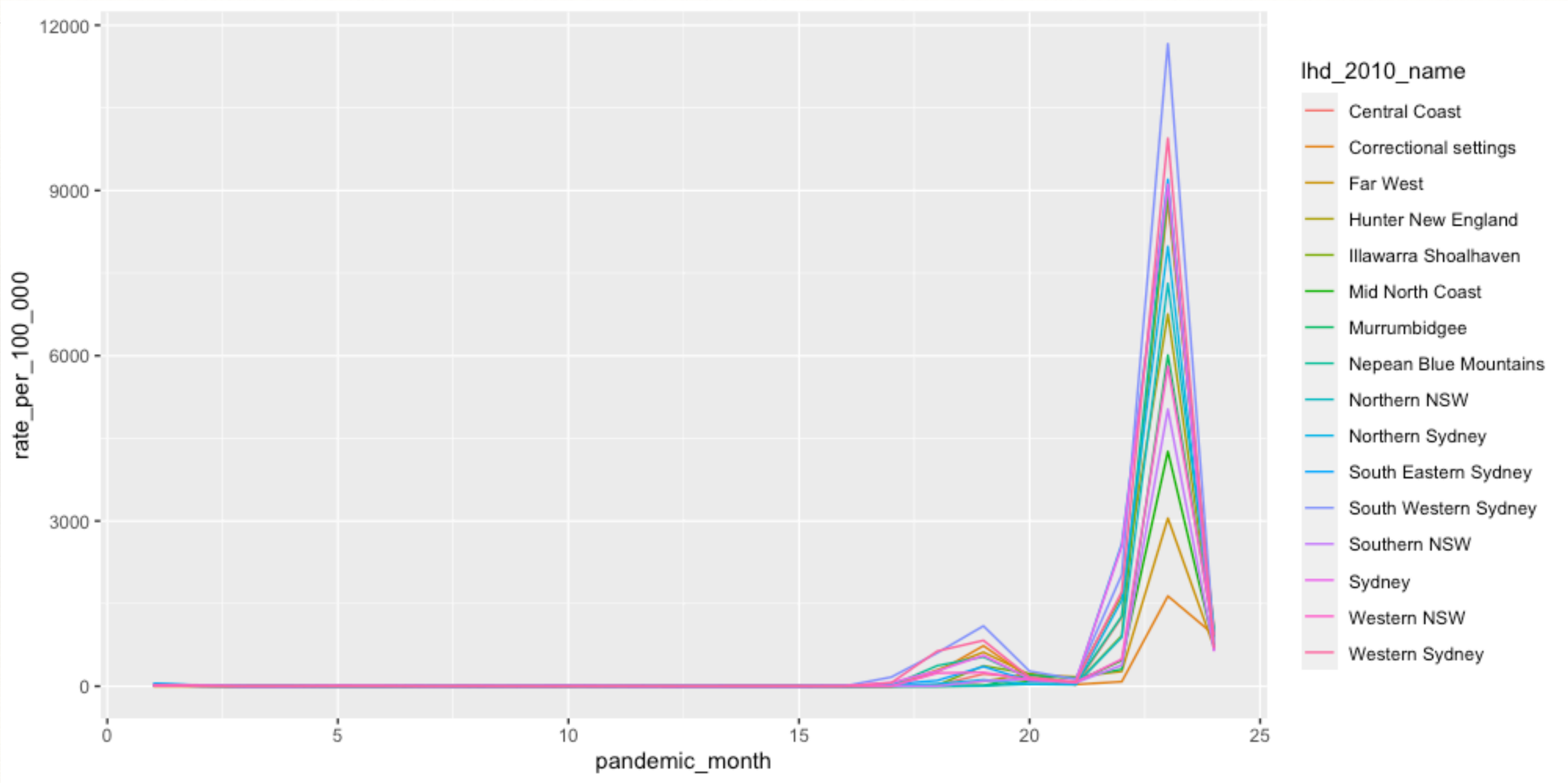
Data: Monthly Covid Cases for each Health District in New South Wales, Australia from March 2020 - February 2022



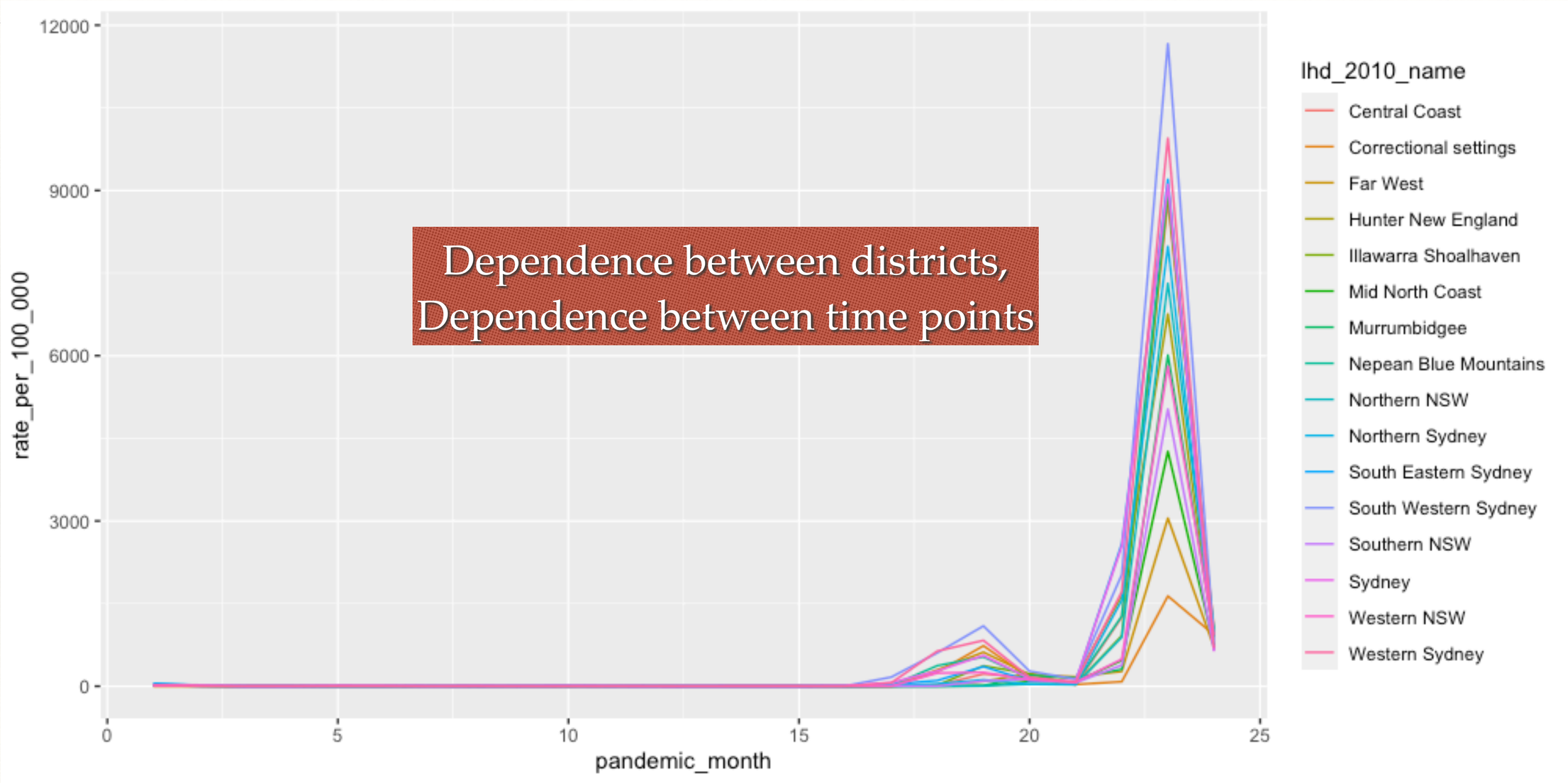
Longitudinal Example

lhd_2010_name	year	month	cases	pop	rate_month	pandemic_month
Central Coast	2020	3	98	309000	0.0003	1
Central Coast	2020	4	19	309000	0.0001	2
Central Coast	2020	5	1	309000	0.0000	3
Central Coast	2020	6	4	309000	0.0000	4
Central Coast	2020	7	2	309000	0.0000	5
Central Coast	2020	8	3	309000	0.0000	6
Central Coast	2020	10	1	309000	0.0000	8
Central Coast	2020	12	5	309000	0.0000	10
Central Coast	2021	1	3	309000	0.0000	11
Central Coast	2021	3	1	309000	0.0000	13
Central Coast	2021	7	9	309000	0.0000	17
Central Coast	2021	8	57	309000	0.0002	18
Central Coast	2021	9	686	309000	0.0022	19
Central Coast	2021	10	505	309000	0.0016	20
Central Coast	2021	11	76	309000	0.0002	21
Central Coast	2021	12	3875	309000	0.0125	22
Central Coast	2022	1	27145	309000	0.0878	23
Central Coast	2022	2	3244	309000	0.0105	24
Correctional settings	2020	7	1	41000	0.0000	5
Correctional settings	2021	8	105	41000	0.0026	18

Longitudinal Example

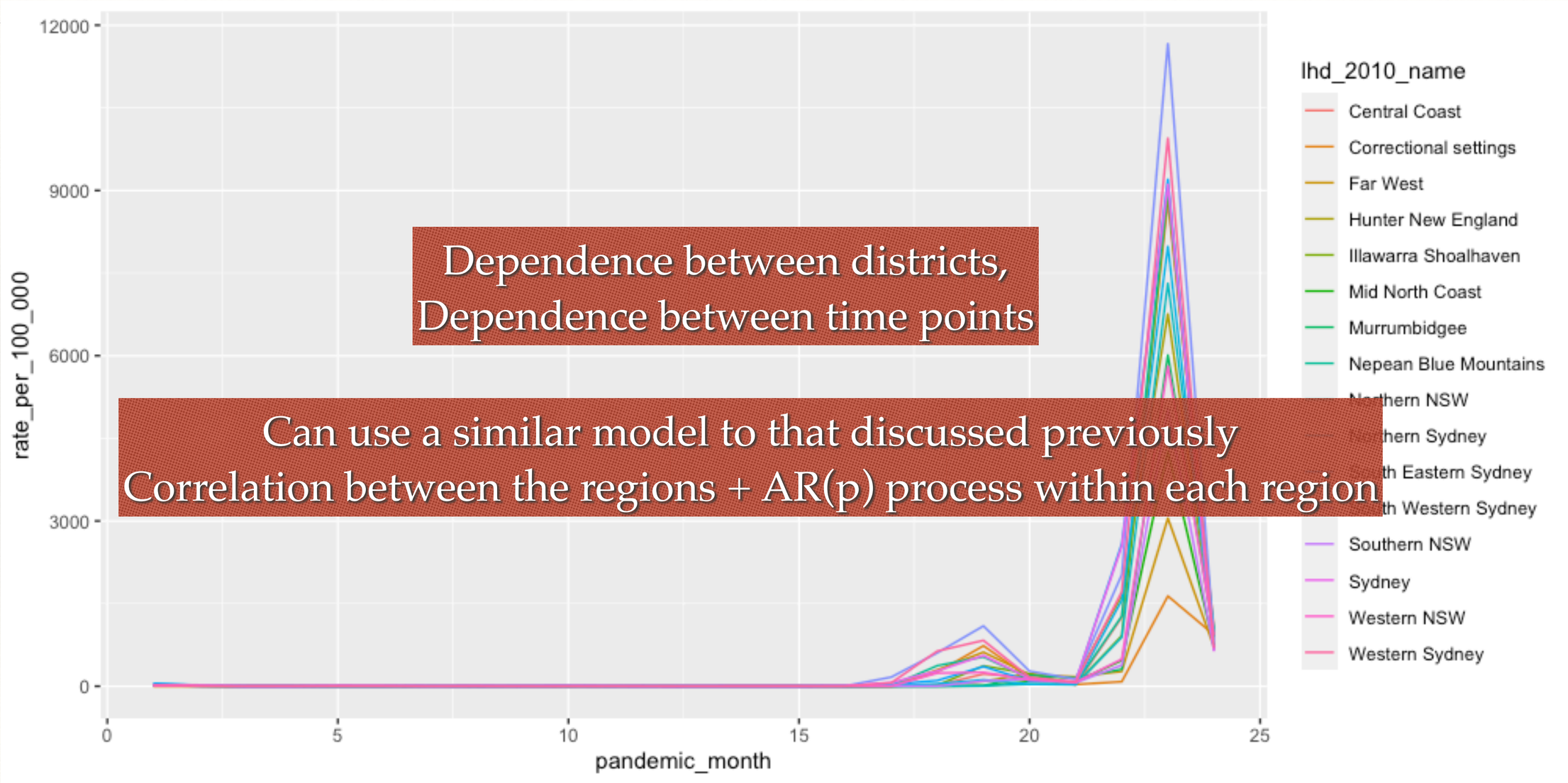


Longitudinal Example



Dependence between districts,
Dependence between time points

Longitudinal Example



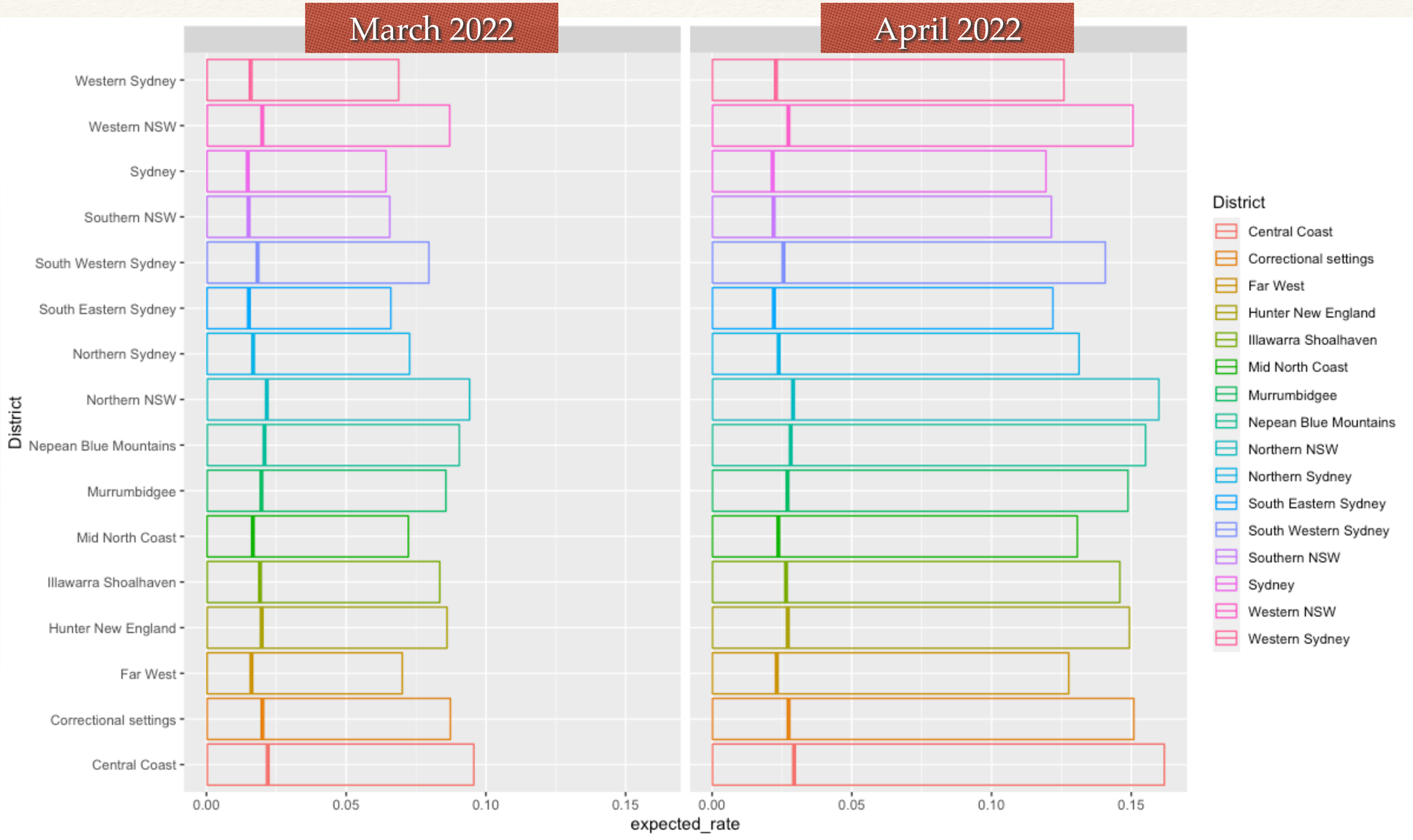
Dependence between districts,
Dependence between time points

Can use a similar model to that discussed previously
Correlation between the regions + AR(p) process within each region

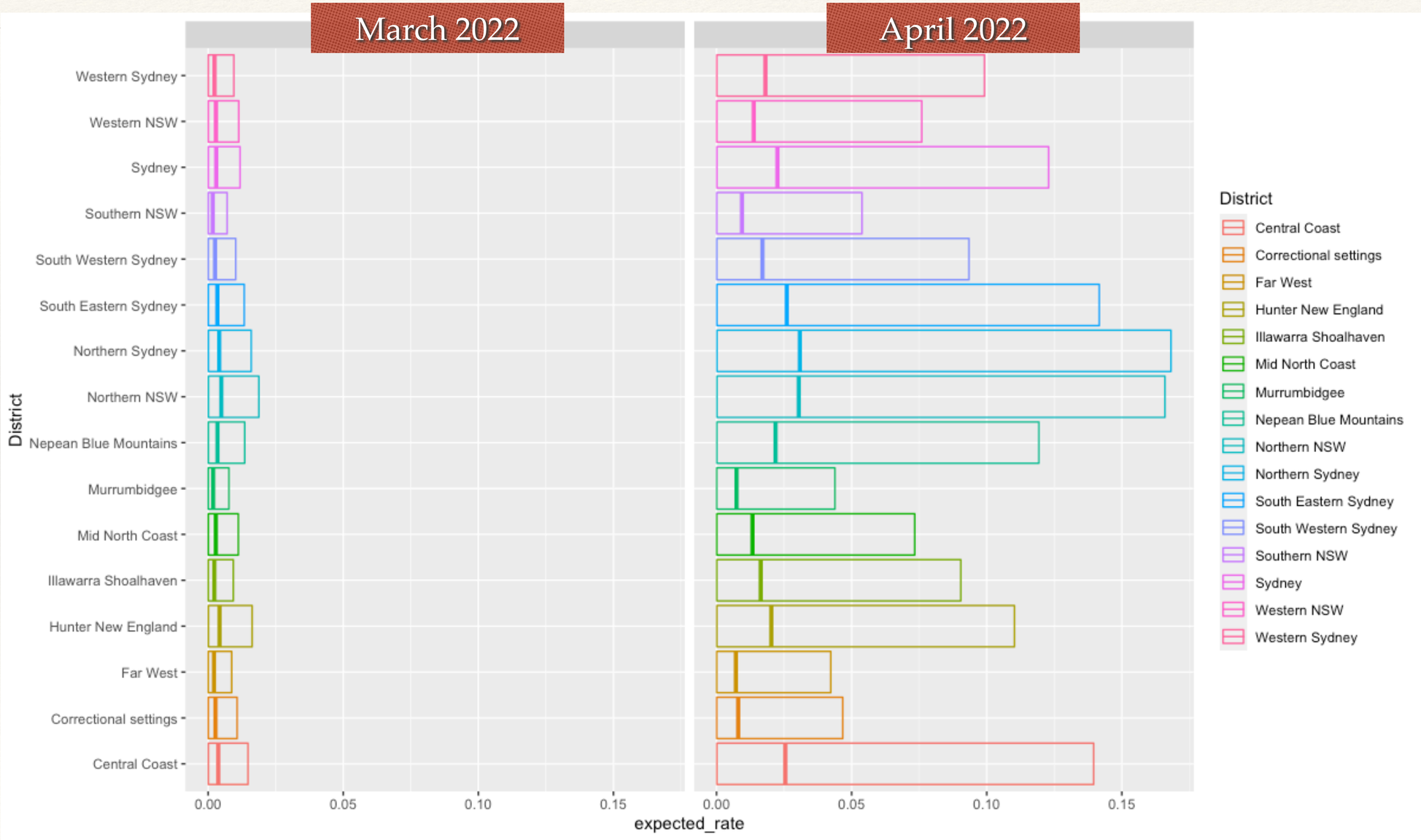
Goal: Predictions for March and April 2022

Compare a model with AR(1) and AR(5)

AR(1) Predictions



AR(5) Predictions



Caveats

These are pretty **naive models** with respect to covid and for **pedagogical purposes only**

How could we improve these models?

1. Spatial Dependence

(Bayesian Spatial Models are good examples of Hierarchical Models, ask Professor Alex Schmidt in the next segment about this)

2. Incorporating more epidemiological information/modelling from Australia + globe more generally

Key Takeaways

1. Bayesian Hierarchical Models allow us to share information between random variables - Both in the likelihood and in the latent space through priors

2. Hierarchies in the prior level can sometimes protect us from poorly calibrated priors

3.There is more to hierarchical priors than random effects + nested models
We can use hierarchies to allow time-varying coefficients, spatially varying coefficients, space-time varying, and really anything else you can think of!!!

More Information

The materials for this workshop can be found at the following GitHub:
https://github.com/tyrelstokes/BayesianHierarchicalWorkshop

I can be found at tyrel.stokes@mail.mcgill.ca

Thanks for attending!