Aalto-yliopisto
Sähkötekniikan
korkeakoulu

Modeling basketball shot likelihoods with Bayesian Binomial-Logit models

Juho Kuikka, Rasmus Siljander

Data

- Player shooting data from years between 2010-2018
- Shot type, Distance from the basket, shot success flag (0 = miss, 1 = shot made)
- Three different shot types
- Shot distances between 15-30

Shot type	Sample size (N)
Jump shot	50000
Fadeaway/Step back	29915
Pullup	50000

How does the distance to the basket affect the likelihood of a successful shot outcome?

- Bernoulli likelihood: $\text{logit}(p_i) = \alpha + \beta x_i$
- Pooled model:

$$p_i \sim binomialLogit(n, \alpha + \beta x_i)$$

$$\alpha = N(0, 10^2)$$

$$\beta = N(0, 10^2)$$

How does the distance to the basket affect the likelihood of a successful shot outcome?

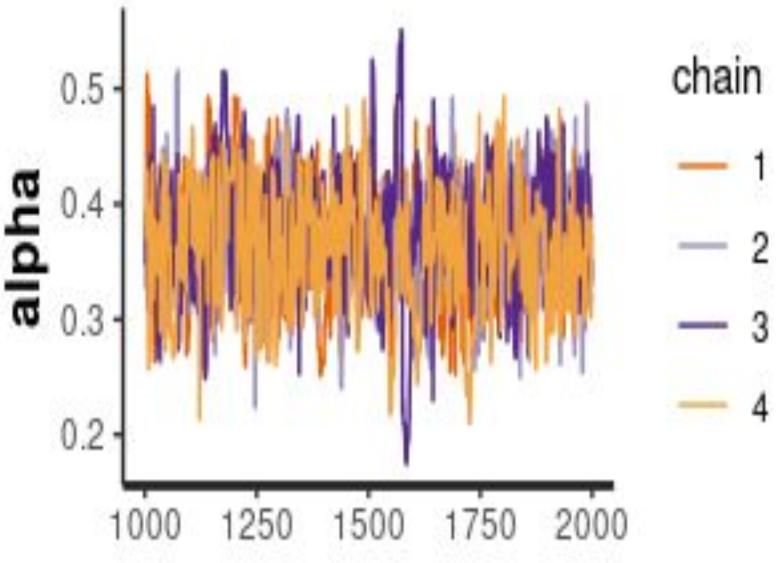
- Hierarchical model:

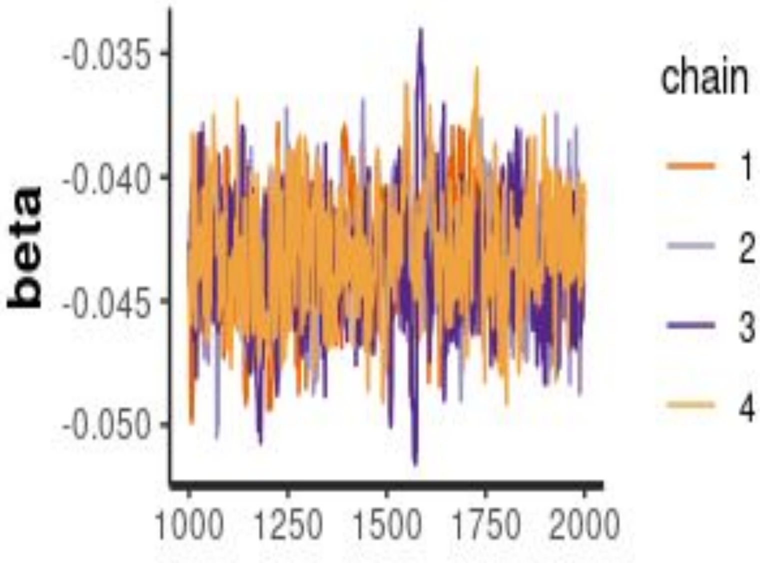
$$p_i \sim binomialLogit(n, \alpha_j + \beta_j x_i)$$

$$\mu_0 \sim N(0, 10^2) \qquad \alpha_j \sim N\left(\mu_0, \sigma^2\right)$$

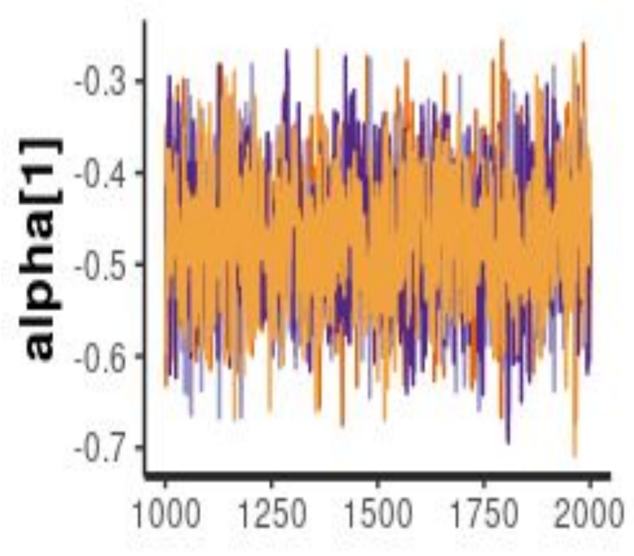
$$\sigma \sim Gamma(1, 1) \qquad \beta_j \sim N\left(\mu_0, \sigma^2\right)$$

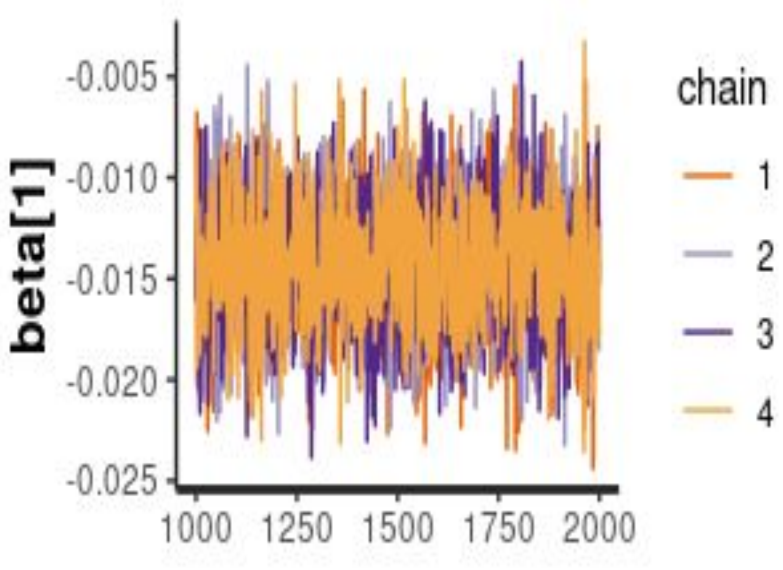
Convergence diagnostics Pooled model



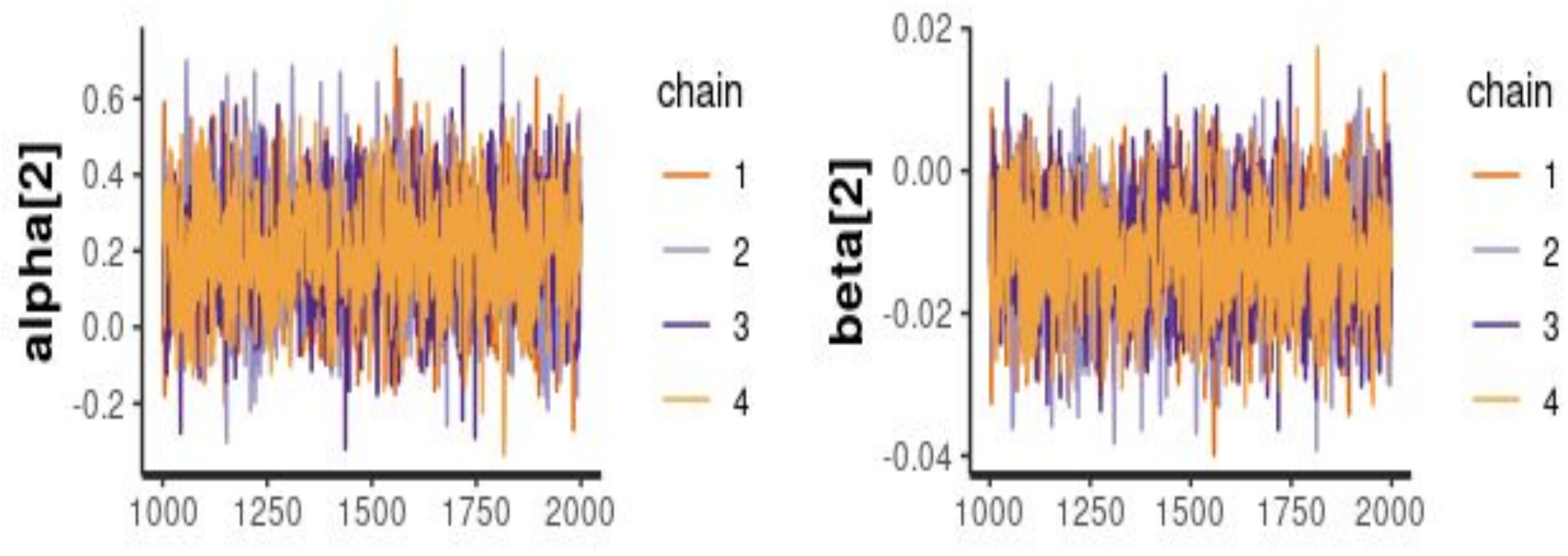


Convergence diagnostics hierarchical

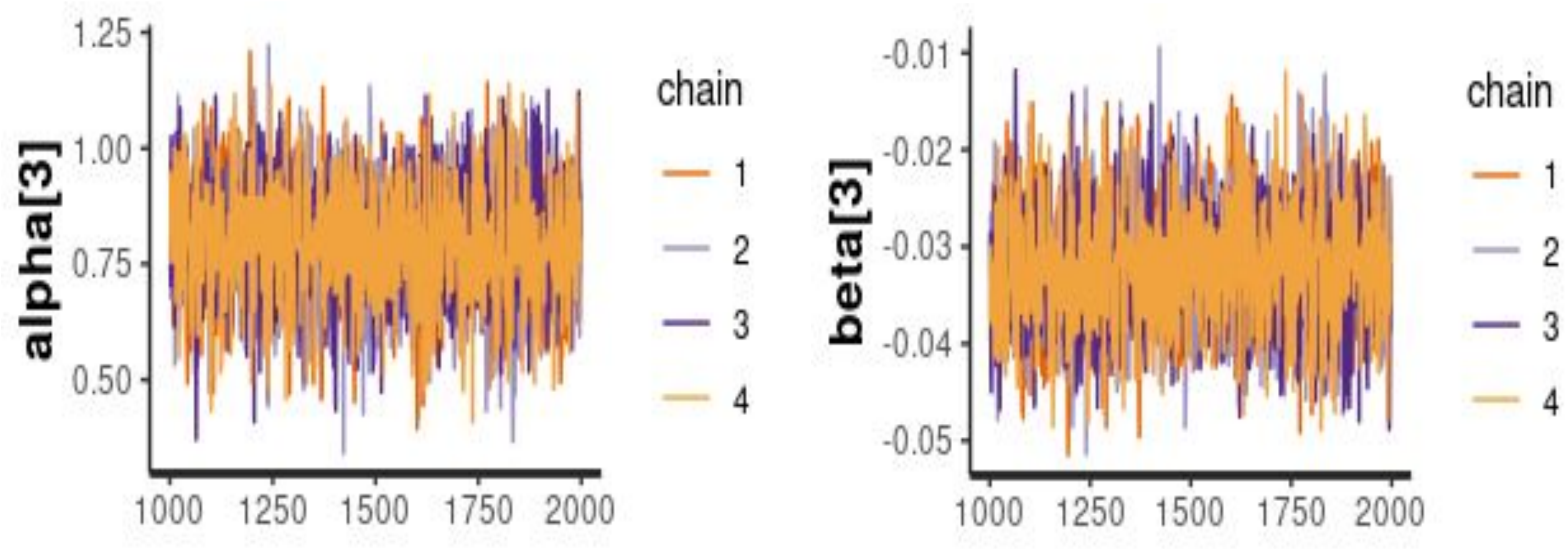




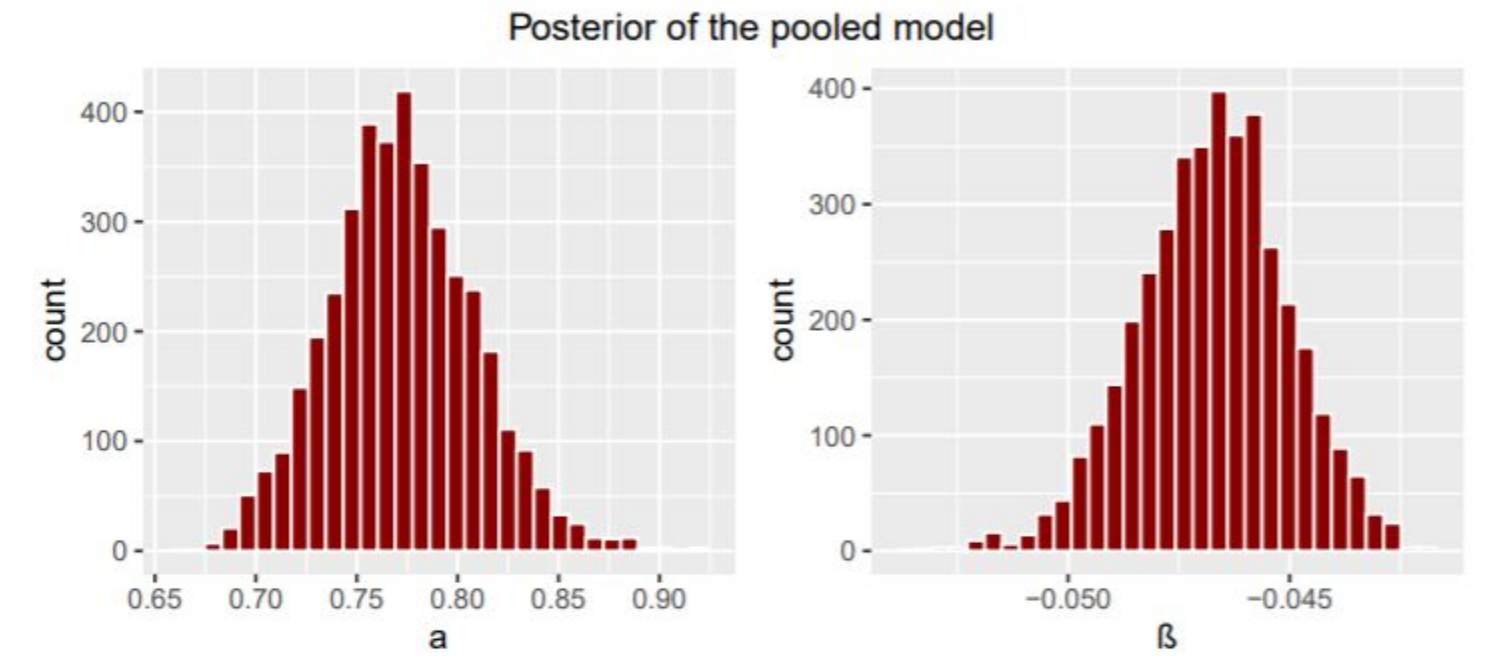
Convergence diagnostics hierarchical



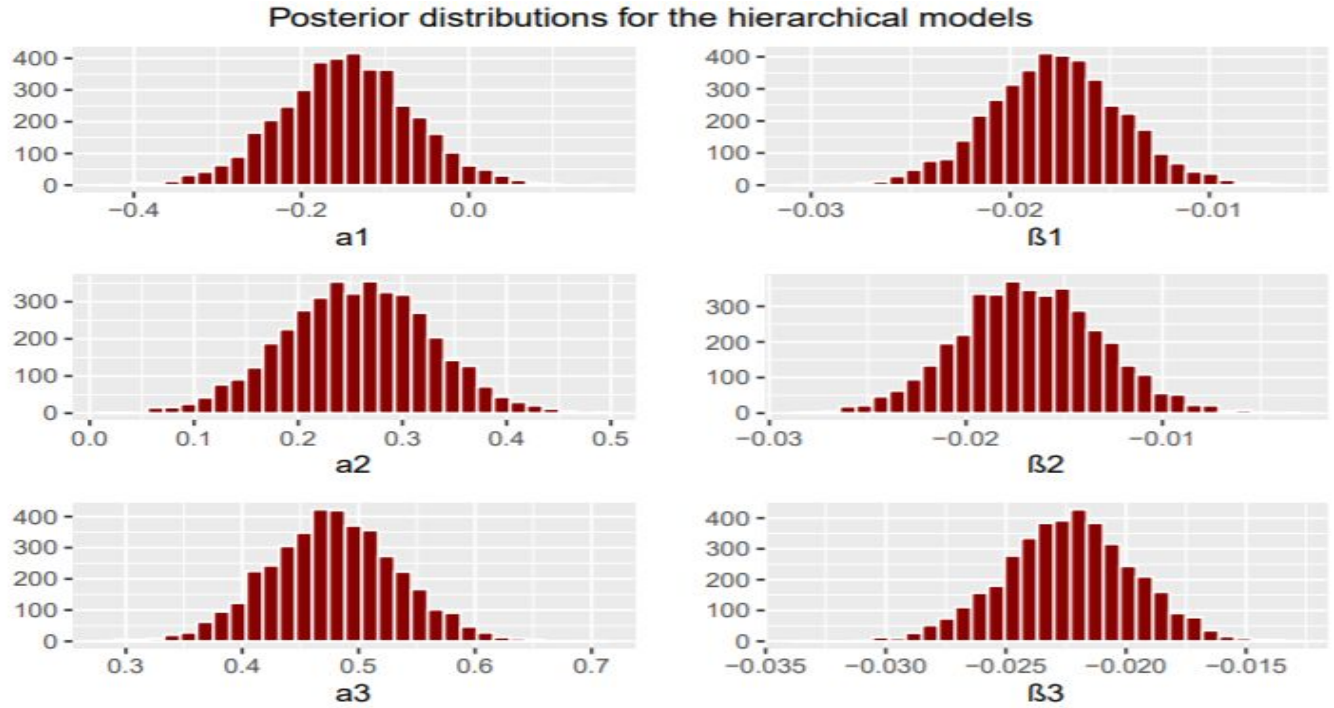
Convergence diagnostics hierarchical



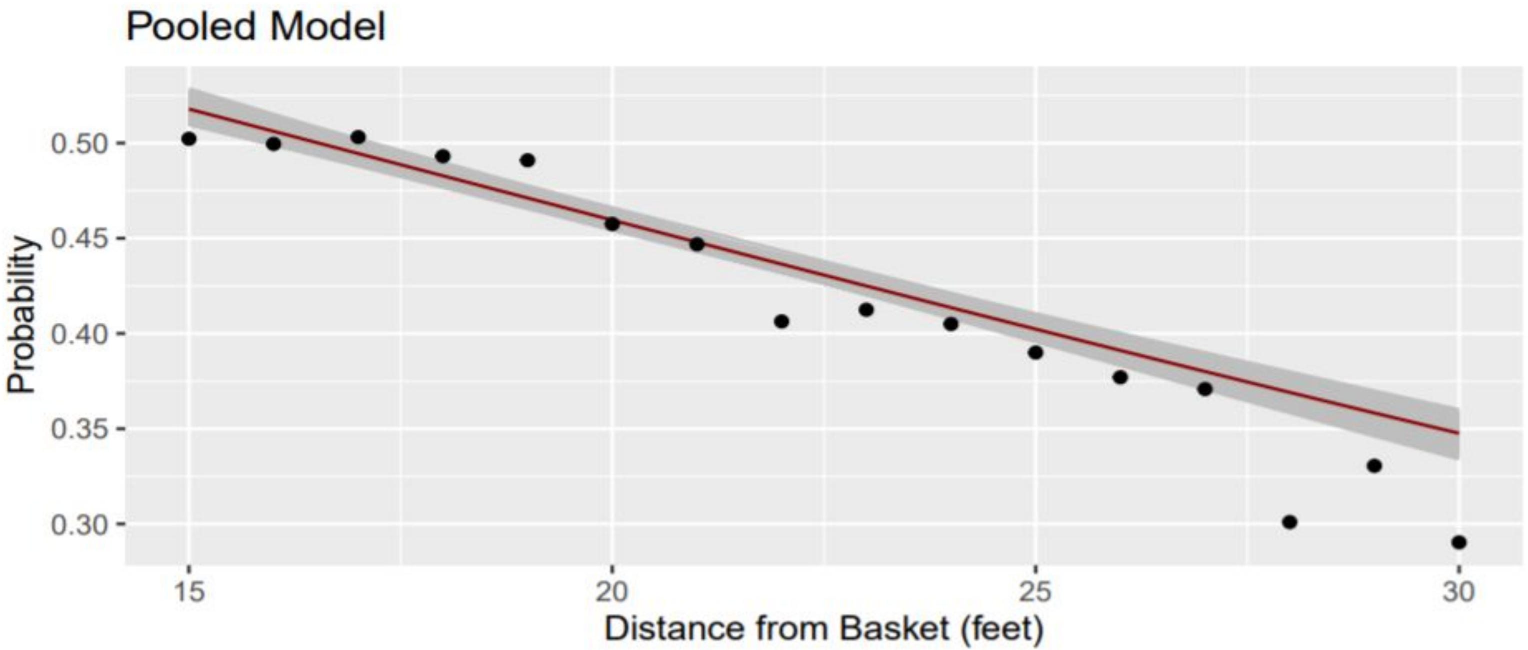
Posterior distributions



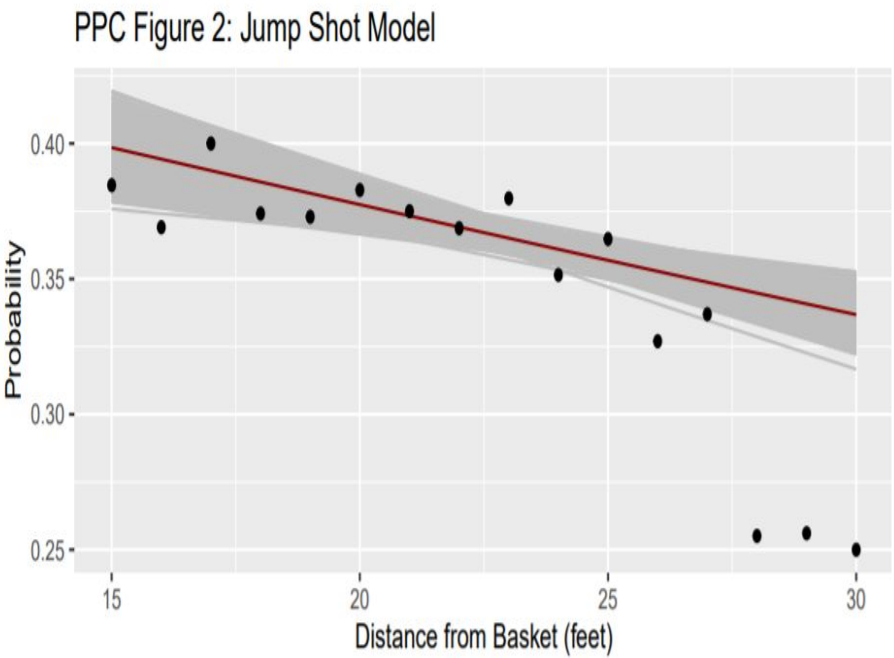
Posterior distributions

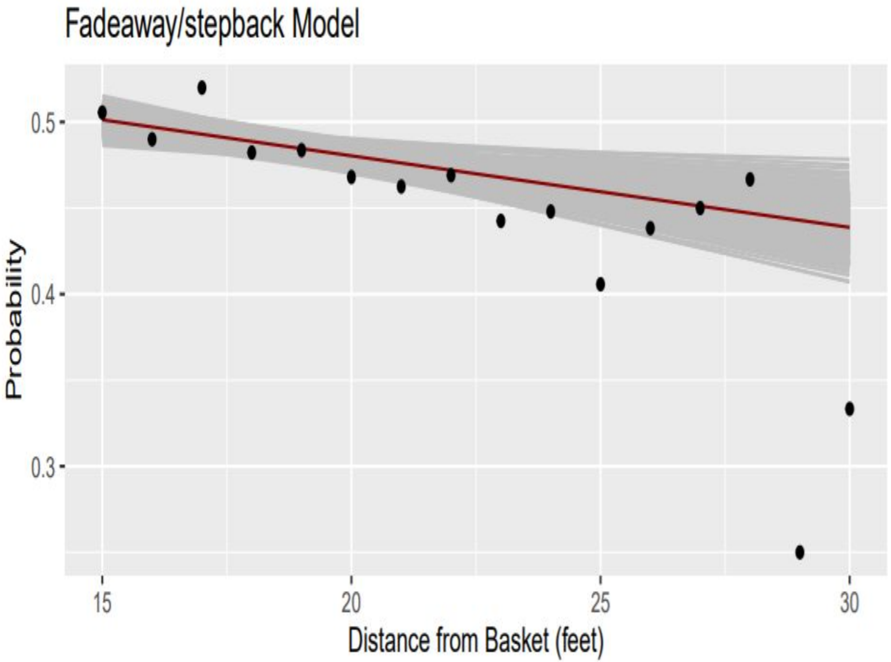


Posterior predictive checking

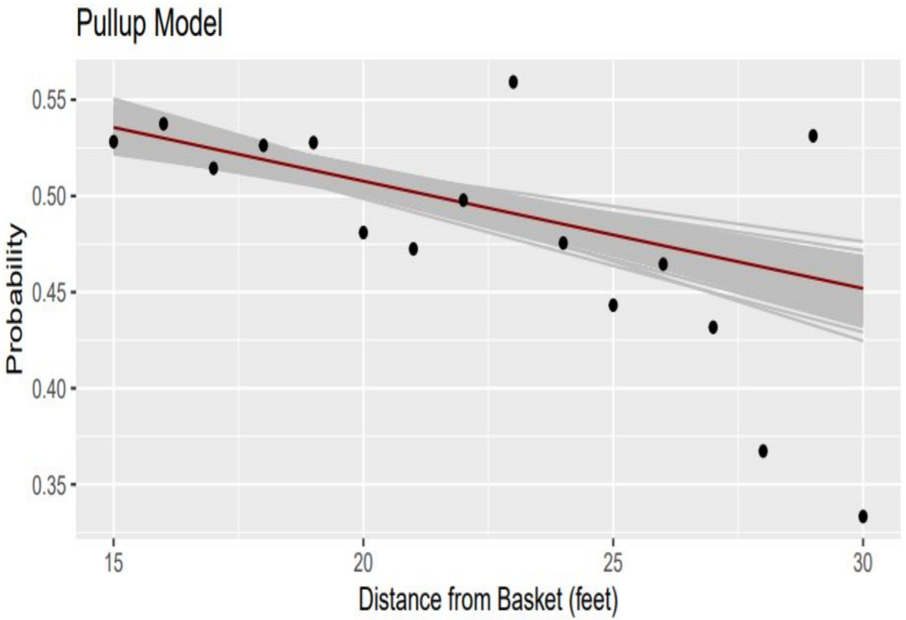


Posterior predictive checking





Posterior predictive checking

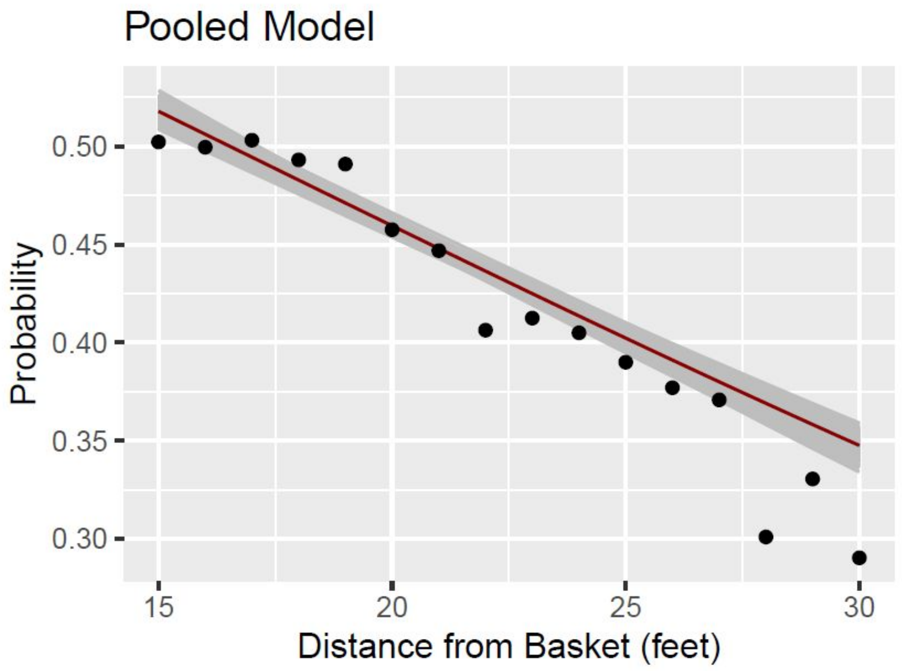


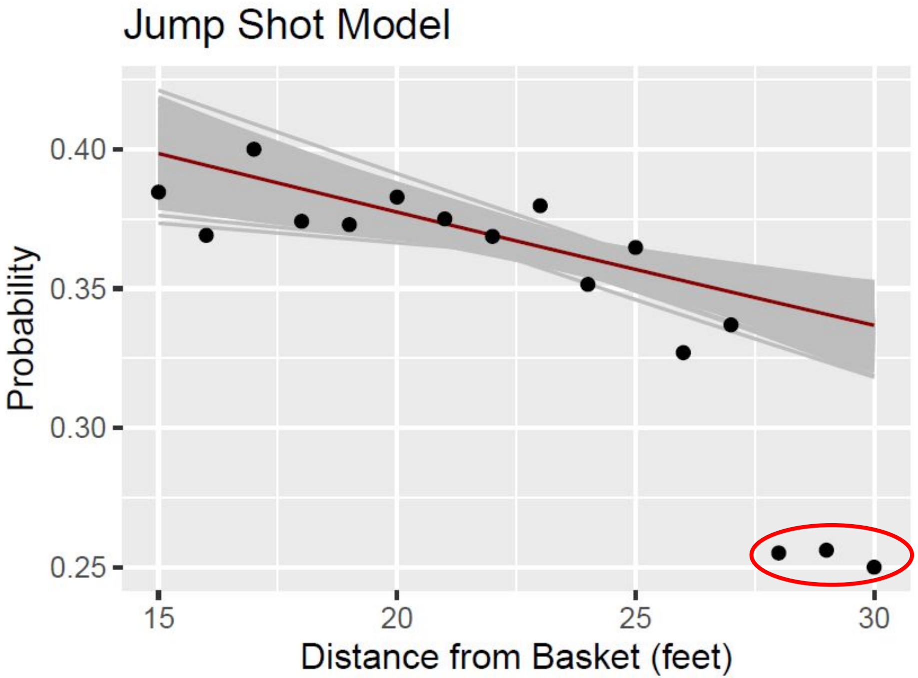
- moderate posterior predictability can be observed

Model comparison and predictive performance

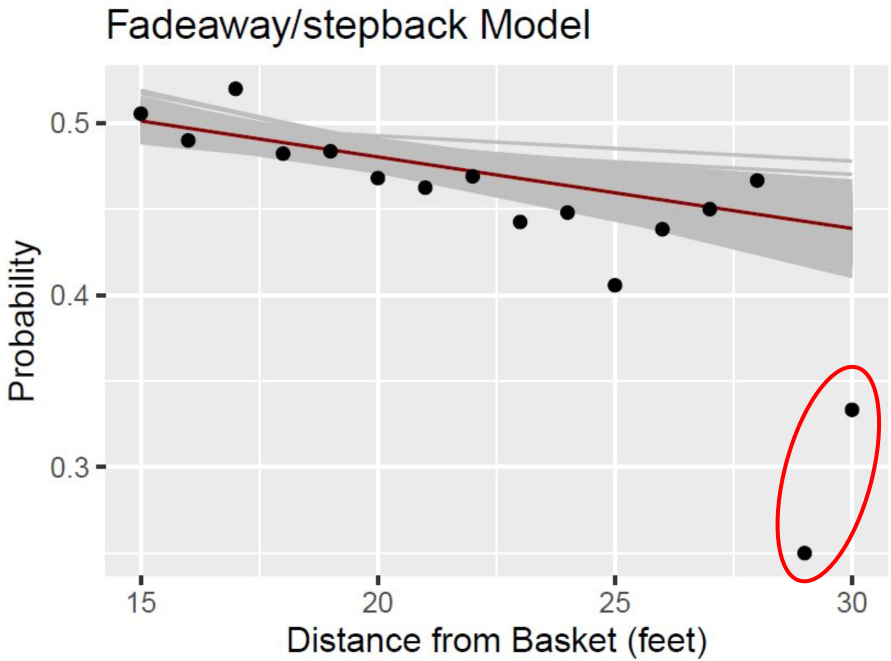
- Visual diagnostics:
  - Posterior predictive plots
  - Residuals

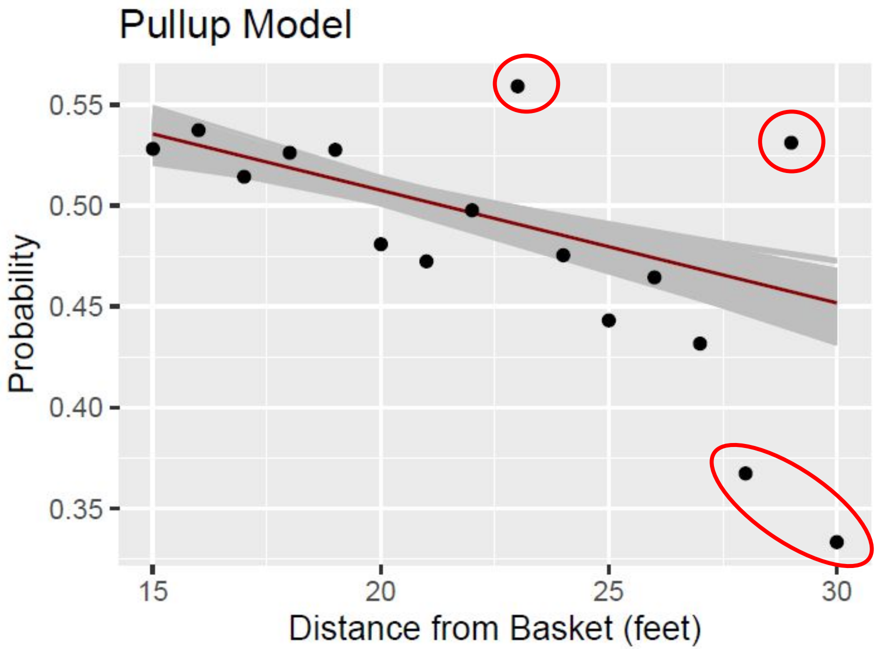
- PSIS LOO diagnostics:
  - Expected log pointwise predictive density (elpd)
  - Pareto k-values


Pooled Model
Probability
0.50
0.45
0.40
0.35
0.30
15
20
25
30
Distance from Basket (feet)

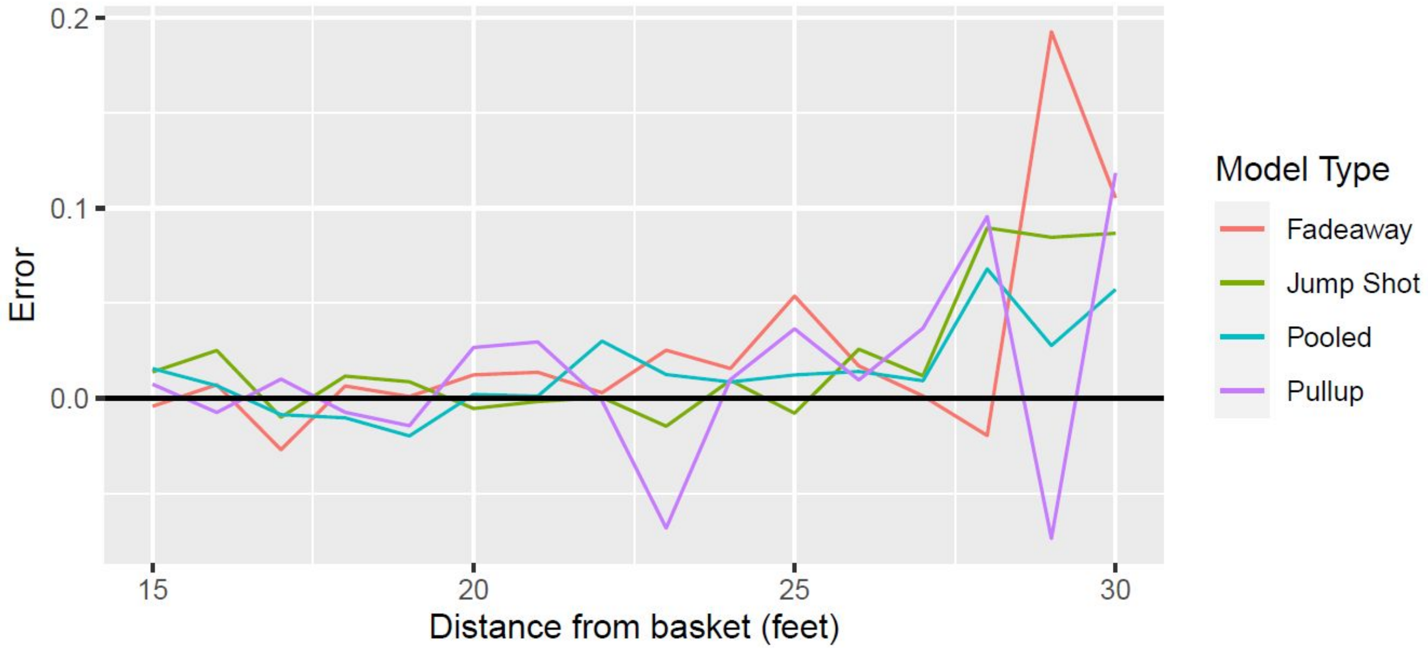

Jump Shot Model
Probability
0.40
0.35
0.30
0.25
15
20
25
30
Distance from Basket (feet)

Predictive Performance





Residuals



0.2
0.1
0.0
Error
15
20
25
30
Distance from basket (feet)
Model Type
Fadeaway
Jump Shot
Pooled
Pullup

PSIS LOO Diagnostics

Pooled Model

```
Computed from 4000 by 16 log-likelihood matrix

         Estimate   SE
elpd_loo    -98.1  8.8
p_loo         6.2  2.3
looic       196.1 17.6
------
Monte Carlo SE of elpd_loo is NA.
```

Hierarchical Model

```
Computed from 4000 by 48 log-likelihood matrix

         Estimate   SE
elpd_loo   -243.7 13.8
p_loo        12.7  2.5
looic       487.4 27.5
------
Monte Carlo SE of elpd_loo is 0.1.
```

PSIS LOO Diagnostics

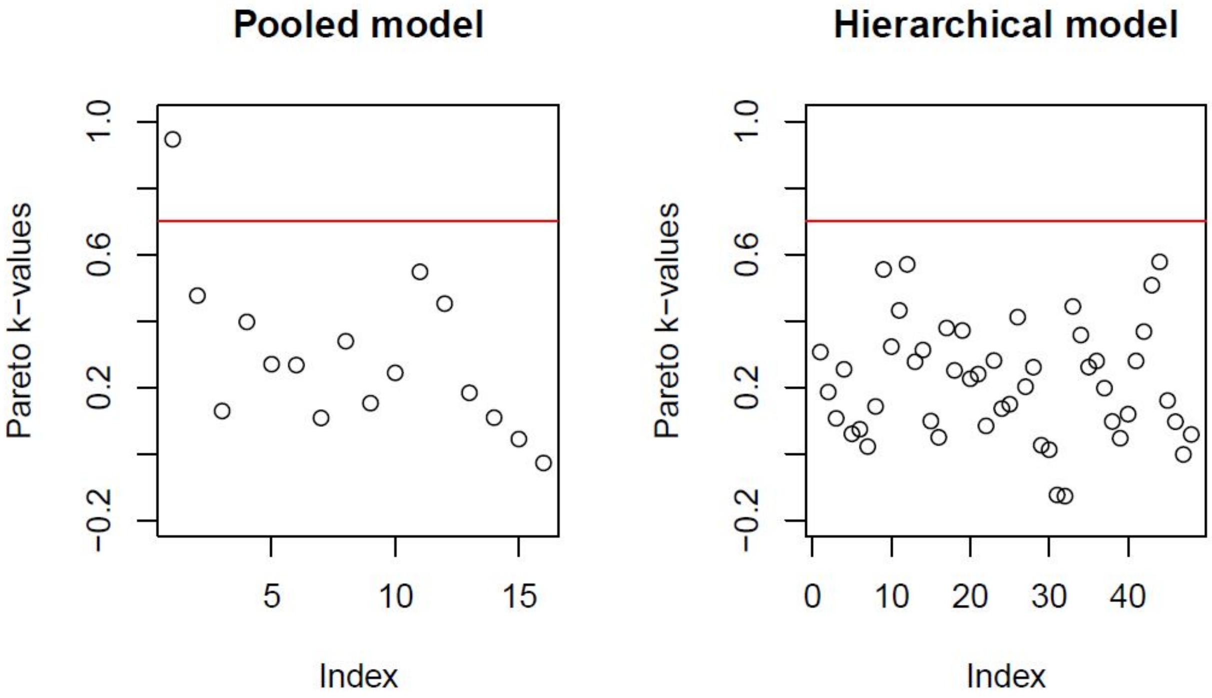
Pooled Model

```
Computed from 4000 by 16 log-likelihood matrix

          Estimate   SE
elpd_loo     -98.1  8.8
p_loo          6.2  2.3
looic        196.1 17.6
------
Monte Carlo SE of elpd_loo is NA.
```

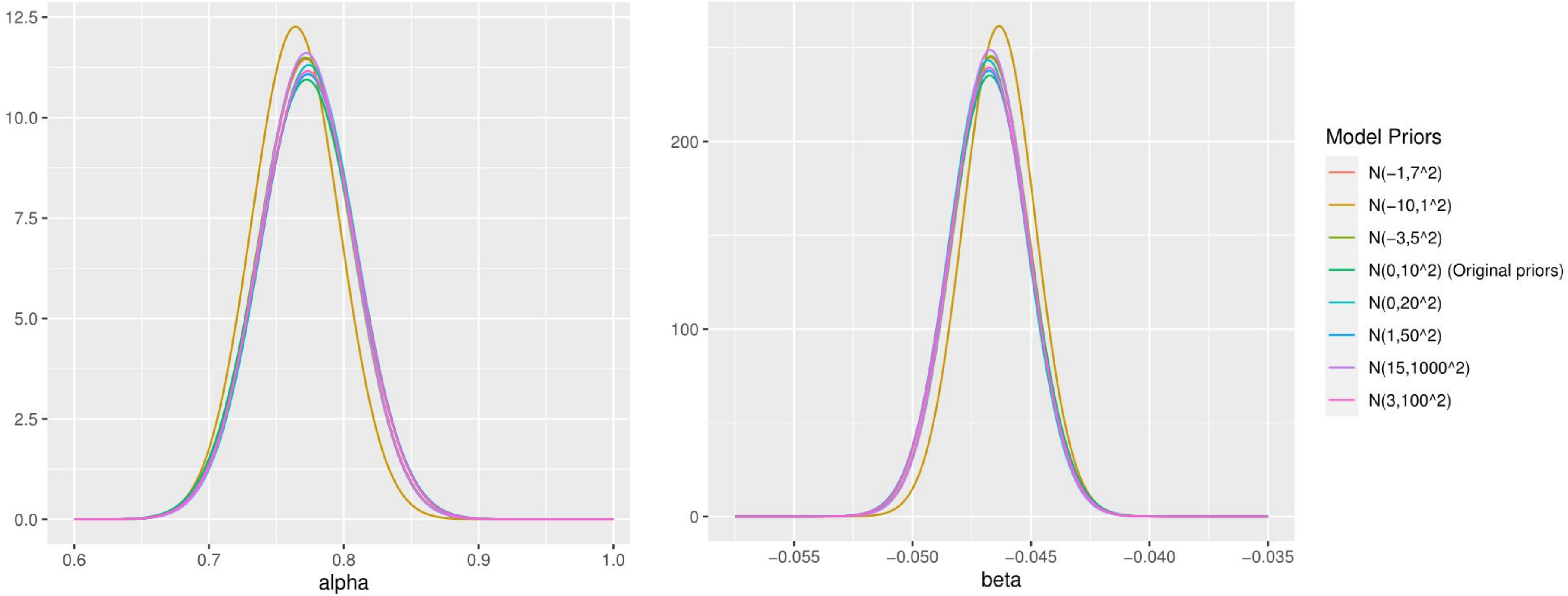
Hierarchical Model

```
Computed from 4000 by 48 log-likelihood matrix

          Estimate   SE
elpd_loo    -243.7 13.8
p_loo         12.7  2.5
looic        487.4 27.5
------
Monte Carlo SE of elpd_loo is 0.1.
```

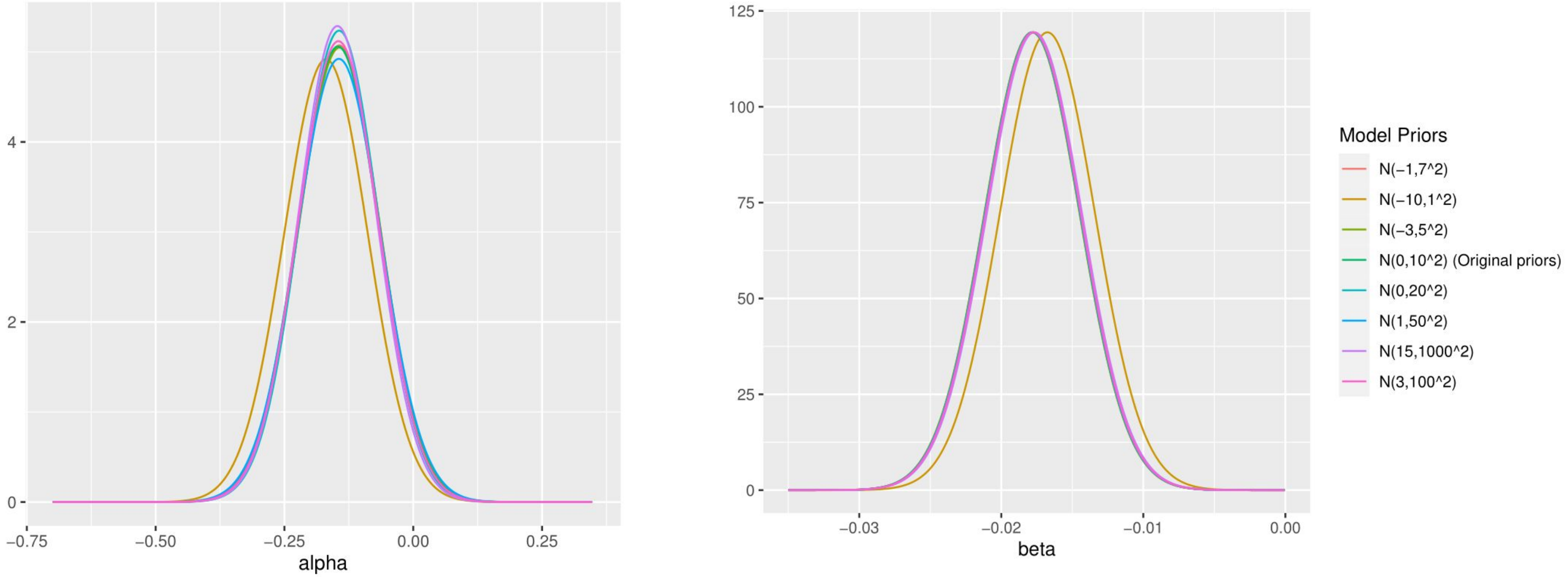
Pareto k-values



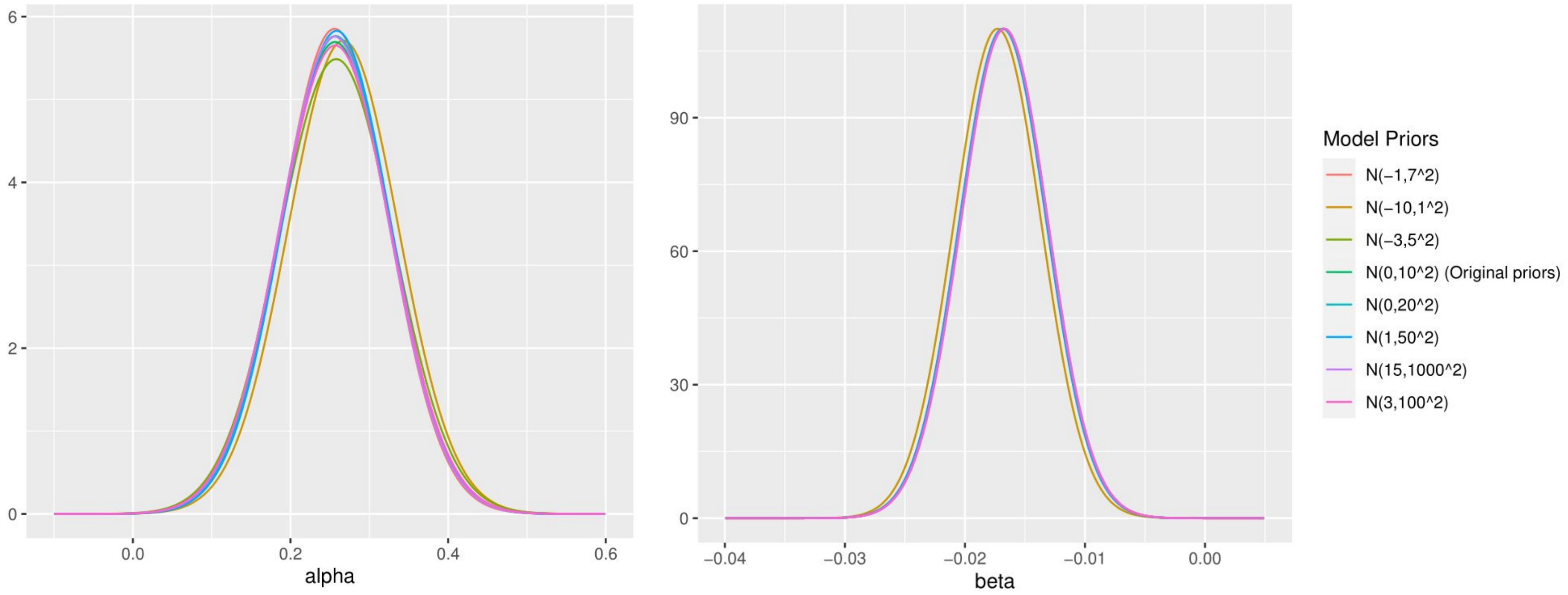
Sensitivity analysis - Pooled Model



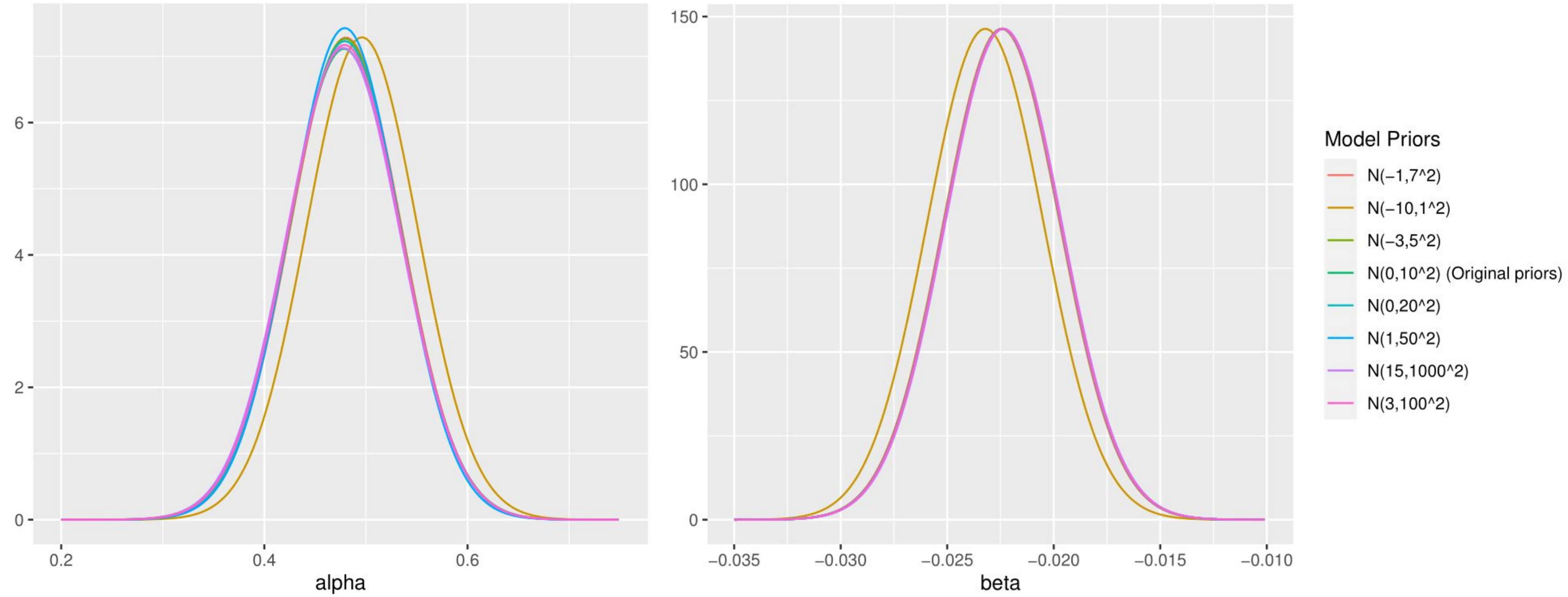
Sensitivity - Jump Shot



Sensitivity - Fadeaway Shot



Sensitivity - Pullup Shot



The pooled model seemed marginally better - but was it?

- Data structure and sampling issues left model comparison less meaningful
  - Pooled model data =/= Hierarchical data

The pooled data could have been partitioned into shot type groups

shot_made_fla	action_type	shot_dista
0	Jump Shot	26
1	Step Back Jump s	17
1	Pullup Jump shot	23
0	Jump Shot	25
1	Pullup Jump shot	24
1	Jump Shot	25
1	Jump Shot	21
0	Pullup Jump shot	25
1	Fadeaway Jump S	26
0	Jump Shot	25
0	Fadeaway Jump S	25
0	Jump Shot	28
0	Fadeaway Jump S	25

Constructed into success ratios,
regardless of shot type

Distance	Throws	Successes
15	n1	s1
16	n2	s2
…	…	…

Size = 16x3

Pooled Model

```
Computed from 4000 by 16 log-likelihood matrix

         Estimate   SE
elpd_loo    -98.1  8.8
p_loo         6.2  2.3
looic       196.1 17.6
------
Monte Carlo SE of elpd_loo is NA.
```

Hierarchical Model

```
Computed from 4000 by 48 log-likelihood matrix

         Estimate   SE
elpd_loo   -243.7 13.8
p_loo        12.7  2.5
looic       487.4 27.5
------
Monte Carlo SE of elpd_loo is 0.1.
```

Conclusion

- Pooled model performed marginally better
- Errors in data set definition mean that meaningful inference is not possible
- Basketball is an extremely complex sport, and univariate modeling may not be plausible at all

- Further reading:

Reich, Brian & Hodges, James & Carlin, Bradley & Reich, Adam. (2006). A Spatial Analysis of Basketball Shot Chart Data. The American Statistician. 60. 3-12. 10.1198/000313006X90305.

Available online here

- Data available at: www.nbasavant.com

```
data {

  int<lower=0> N;                // Number of shot distances
  vector[N] distances;           // Vector of shot distances
  int throws[N];                 // Throws per shot distance
  int successes[N];              // Number of successes per shot distance pair

  real mu0_alpha;                // Priors
  real sigma0_alpha;

  real mu0_beta;
  real sigma0_beta;

}

parameters {
  real alpha;                     // Model parameters
  real beta;
}

transformed parameters {
  vector[N] logit_p = alpha + beta * distances;
}

model {
  alpha ~ normal(mu0_alpha, sigma0_alpha);        // Weakly informative priors
  beta ~ normal(mu0_beta, sigma0_beta);

  successes ~ binomial_logit(throws, logit_p);
}


generated quantities {
  vector[N] log_lik;
  for (i in 1:N){
    log_lik[i] = binomial_logit_lpmf(successes[i]| throws[i], logit_p[i]);
  }
}
```

```
data {
  int<lower=0> N;                  // number of shot distances per group
  int<lower=0> J;                  // number of shot types

  vector[N] distances;             // Vector of shot distances
  int throws[N,J];                 // Throws per shot distance and shot type
  int successes[N,J];              // Throws per shot distance and shot type

  real mu0_hyper;                // Priors
  real sigma0_hyper;

}

parameters {
  real mu0;                        // prior mean
  real<lower=0> sigma;             // prior std (constrained to be positive)
  vector[J] alpha;                 // shot type alpha
  vector[J] beta;                  // shot type beta
}

model {
  mu0 ~ normal(mu0_hyper,sigma0_hyper);     // weakly informative prior
  sigma ~ gamma(1,1);                       // weakly informative prior

  for (i in 1:J) {
    alpha[i] ~ normal(mu0, sigma);  //parameters are modeled from group-specific distributions
    beta[i] ~ normal(mu0, sigma);
    successes[,i] ~ binomial_logit(throws[,i], alpha[i] + beta[i]*distances);
  }
}

generated quantities {
  vector[N*J] log_lik;
  for (j in 1:J) {
    for (i in 1:N) {
      log_lik[(j-1)*N + i] = binomial_logit_lpmf(successes[i,j]| throws[i,j],
      alpha[j] + beta[j]*distances[i]);
    }
  }
}
```

Stan Code Error

$$p_{ij} \sim BinomialLogit(n, \alpha_j + \beta_j x_i)$$

$$\alpha_j \sim N(\mu_0, \sigma^2)$$

$$\beta_j \sim N(\mu_0, \sigma^2)$$

$$\mu_0 \sim N(0, 10^2)$$

$$\sigma \sim Gamma(1, 1)$$

```
model {
  mu0 ~ normal(mu0_hyper,sigma0_hyper);      //
  sigma ~ gamma(1,1);                        //

  for (i in 1:J) {
    alpha[i] ~ normal(mu0, sigma);  //parameter
    beta[i] ~ normal(mu0, sigma);
    successes[,i] ~ binomial_logit(throws[,i],
  }
}
```

Stan Code Error

$$p_{ij} \sim BinomialLogit(n, \alpha_j + \beta_j x_i)$$
$$\alpha_j \sim N(\mu_0, \sigma^2)$$
$$\beta_j \sim N(\mu_0, \sigma^2)$$
$$\mu_0 \sim N(0, 10^2)$$
$$\sigma \sim Gamma(1, 1)$$

```
model {
  mu0 ~ normal(mu0_hyper,sigma0_hyper);      //
  sigma ~ gamma(1,1);                        //

  for (i in 1:J) {
    alpha[i] ~ normal(mu0, sigma);  //paramete
    beta[i] ~ normal(mu0, sigma);
    successes[,i] ~ binomial_logit(throws[,i],
  }
}
```