

# Field Goal Kicking using Bayesian Stats

Collin Cook

## Setup

This project is modified from an assignment in a Bayesian analysis course at Ohio State.

To begin, we select a pre-determined sample of kickers from our data, with variables indicating the kicker (ID's from 1-25), a binary made/miss indicator, and the distance where the kick was attempted from.

```
# Import Full data
data <- read_excel("FG_Kicks.xlsx")

# Selection of kickers
# 22 3 13 25 4 10 24 1 20
players <- c(22, 3, 13, 25, 4, 10, 24, 1, 20)
kicks <- data %>% filter(Player %in% players)

# Preview
knitr::kable(head(kicks, 5))
```

Player	Made	Distance
1	1	33
1	1	35
1	1	48
1	1	45
1	0	36

```
# Attempts per kicker
knitr::kable(table(kicks$Player))
```

Var1	Freq
1	32
3	30
4	33
10	34
13	30
20	33
22	32
24	31
25	32

Next, we will use this data to run a Bayesian Hierarchical model to predict the probability of success of a made kick at any distance for each kicker. This is done use the JAGS specifications in R.

## The Model

We wish to model the probability of success for a kick at some given distance, knowing that we have multiple kickers with different rates of success. We start by assuming that the outcome of the kick for each kicker is conditionally independent

$$p(\text{Made}|\alpha, \beta) = \prod_{j=1}^{N_P} \prod_{i=1}^{N_K} p(\text{Made}_{ij}|\alpha_j, \beta_j)$$

We can model the outcome of the kicks as

$$\text{Made}_{ij} \sim \text{Bernoulli}(\theta_{ij})$$

In this setup, we have distinctions of individual kick attempt  $i = 1 \dots N_K$  for each of the  $j = 1 \dots N_P = 9$  kickers in our data. Now, we can model the probability  $\theta$  as a logistic regression where the probability of a made field goal depends on distance, and is different for each kicker.

$$\text{logit}(\theta_{ij}) = \alpha_j + \beta_j \cdot \text{Distance}_{ij}$$

For this model, we expect negative values for  $\beta$  to model how probability goes down as distance increases. In this case,  $\alpha_j$  is the baseline log-odds of making a field goal, and  $\beta_j$  is the decrease in log-odds for each additional yard.

Next, we assume conditional independence for  $\alpha, \beta$

$$p(\alpha, \beta | \mu_\alpha, \mu_\beta, \sigma_\alpha, \sigma_\beta) = \prod_{j=1}^{N_P} p(\alpha_j | \mu_\alpha, \sigma_\alpha) p(\beta_j | \mu_\beta, \sigma_\beta)$$

Since we expect negative values for some coefficients, we can model them as

$$\alpha_j \sim \text{Normal}(\mu_\alpha, \sigma_\alpha) \quad \beta_j \sim \text{Normal}(\mu_\beta, \sigma_\beta)$$

Now, the  $\mu$  values track the mean baseline log-odds and distance coefficient for each kicker, and the  $\sigma$  values track their standard deviation. This also allows us to provide an informed initial guess and range for the model to explore. In addition to the kick data given, we know that ~98% of kicks are made around 25 yards, and drops to ~60% when pushed to around 55 yards. Using these points, the following logistics estimate was generated:

$$\text{logit}(\theta_{\text{prior}}) \approx 3 - \frac{5.125}{100} \text{Distance}$$

Therefore, we set priors of

$$\mu_\alpha \sim \text{Normal}(3, 1) \quad \text{and} \quad \mu_\beta \sim \text{Normal}(0.05, 0.01)$$

and guess

$$\sigma_\alpha \sim \text{Normal}(1, 0.01) \quad \text{and} \quad \sigma_\beta \sim \text{Normal}(1, 0.01)$$

## Coding the Model

```
# JAGS Code
# Setup -----
n <- length(kicks$Made)
# Create new Player ID, ordered 1-9
```

```

kicks$ID <- rep(NA, n)
for (i in 1:n){
  kicks$ID[i] <- which(players == kicks$Player[i])
}
NPlayers <- length(players)

# Data list
mydata <- list(n=n, NPlayers=NPlayers, Made=kicks$Made, Distance=kicks$Distance,
              ID=kicks$ID)
# Parameter Initialization - use prior knowledge and exact line fit estimates
myinit <- list(mu.a=3, mu.b=5.125/100,
              sigma.a=0.1, sigma.b=0.05)

# Iteration Control -----
niters <- 50000 # Full amount
nbruns <- 5000 # Burn-in
nadapt <- 1000
nchains <- 2 # Number of chains

# JAGS Model -----
mod <- "model {
  # Likelihood
  for (i in 1:n) {
    Made[i] ~ dbern(theta[i])
    logit(theta[i]) = alpha[ID[i]] + beta[ID[i]]*Distance[i]
  }

  # Priors
  for (j in 1:NPlayers) {
    alpha[j] ~ dnorm(mu.a, sigma.a)
    beta[j] ~ dnorm(mu.b, sigma.b)
  }
  mu.a ~ dnorm(3, 1)
  mu.b ~ dnorm(0.05, 0.01)
  sigma.a ~ dnorm(1, 0.01)
  sigma.b ~ dnorm(1, 0.01)
}"

set.seed(1730)
# Initialize model -----
fit <- jags.model(textConnection(mod), data=mydata, inits=myinit,
                 n.chains=nchains, n.adapt=nadapt)

```

```

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 287
##   Unobserved stochastic nodes: 22
##   Total graph size: 1456
##
## Initializing model

```

```
# Sample from model -----
fit.samples <- coda.samples(fit, c("alpha", "beta", "mu.a", "mu.b",
                                   "sigma.a", "sigma.b"), n.iter=niters)
```

## Results

The above model was fit in R using JAGS to assess the hierarchical nature of the data. Using 50,000 iterations with 5,000 burn-in iterations, 2 chains were run the check for convergence. Below are the summary statistics for the modeled variables for each kicker.

```
# Summary Statistics from JAGS
summary(fit.samples)[1]
```

```
## $statistics
##           Mean          SD      Naive SE Time-series SE
## alpha[1]  4.19475487 0.84299977 2.665799e-03  0.0207388854
## alpha[2]  4.40227533 0.82880699 2.620918e-03  0.0204640543
## alpha[3]  4.54684533 0.82878595 2.620851e-03  0.0221924775
## alpha[4]  4.44047763 0.82974693 2.623890e-03  0.0218259849
## alpha[5]  4.43046624 0.78498751 2.482348e-03  0.0193895437
## alpha[6]  4.51193394 0.82306387 2.602756e-03  0.0219212156
## alpha[7]  4.38355778 0.79176974 2.503796e-03  0.0199063017
## alpha[8]  4.25196371 0.84217900 2.663204e-03  0.0201185035
## alpha[9]  4.45094368 0.82127141 2.597088e-03  0.0202758427
## beta[1]   -0.05471816 0.02281768 7.215584e-05  0.0004696786
## beta[2]   -0.05853594 0.02259961 7.146624e-05  0.0004781849
## beta[3]   -0.08062551 0.02368310 7.489253e-05  0.0005326544
## beta[4]   -0.06073059 0.02049889 6.482317e-05  0.0004593509
## beta[5]   -0.07859013 0.01981072 6.264701e-05  0.0004333259
## beta[6]   -0.07859549 0.01988932 6.289556e-05  0.0004852430
## beta[7]   -0.07496445 0.02021559 6.392732e-05  0.0004393760
## beta[8]   -0.05653728 0.02242258 7.090643e-05  0.0004774548
## beta[9]   -0.06340859 0.02077572 6.569859e-05  0.0004468144
## mu.a      4.35349525 0.63977549 2.023148e-03  0.0200183127
## mu.b     -0.06735654 0.07727688 2.443710e-04  0.0004520227
## sigma.a    8.33733006 6.14784630 1.944120e-02  0.0782893021
## sigma.b   21.69987256 6.96234207 2.201686e-02  0.0298319750
```

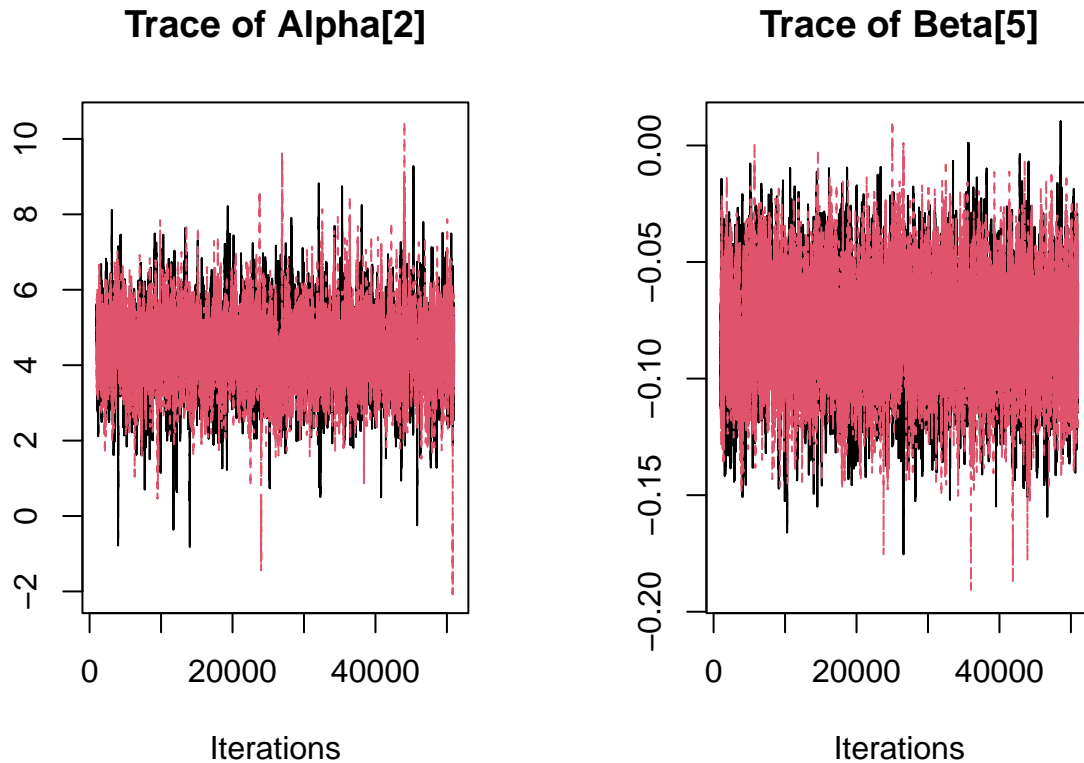
With starting values of  $\mu_\alpha = 3, \mu_\beta = 0.05125, \sigma_\alpha = 0.1, \sigma_\beta = 0.05$ , we get different values that still make sense in terms of the model, especially when plotted out (shown in player evaluations). Additionally, we can interpret the  $\mu_\alpha = 5.31$  and  $\mu_\beta = -0.087$  as the league average, and all of the other kickers vary around that baseline.

In the context of kicking field goals, the  $\alpha$  coefficients represent the baseline log-odds of successful kick attempts, and these values are all about the same, varying around 5.31. On the other hand, the  $\beta$  values vary quite a bit more, showing how variable some kickers are to increased distance. A more extreme distance coefficient means that the logistic regression line becomes steeper, while a value closer to 0 means there is a higher probability of success at higher distances.

The  $\sigma$  values represent the deviation in the distributions of  $\alpha$  and  $\beta$ , and it seems that the estimates are quite a ways off from what we initially thought. It appears that this area of the model may not be as accurate as other parameters, but we aren't too interested in these values anyways.

Next, a select few traceplots are shown, to show convergence of the chains to reasonable values over the many iterations.

```
par(mfrow=c(1,2))
# Trace plots
# Alpha 3
traceplot(fit.samples[,2], main="Trace of Alpha[2]")
traceplot(fit.samples[,14], main="Trace of Beta[5]")
```



For the plots, we can see that they explore reasonable values but converge after many iterations. And for the summary statistics, we can see similar estimated standard deviations for most of the coefficients, besides the  $\sigma$  values. It looks like we have similar yet distinguishable  $\alpha$  and  $\beta$  values showing the differences between each kicker.

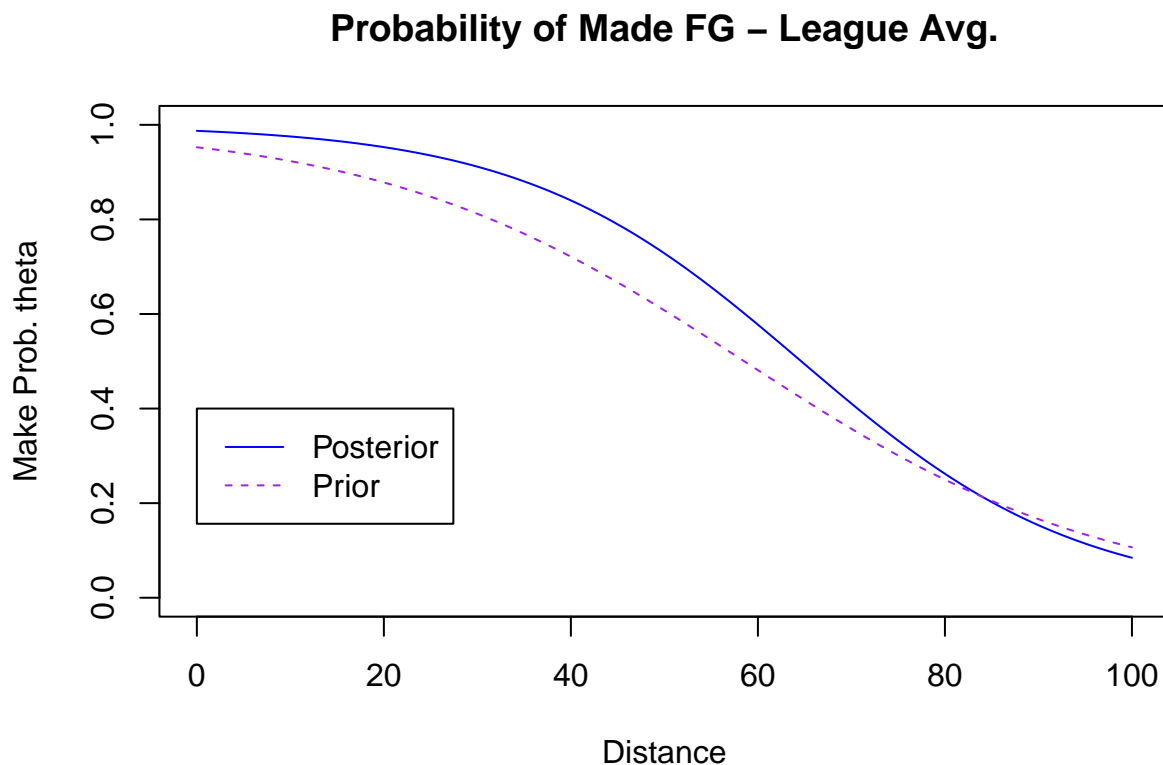
Using the  $\mu_\alpha, \mu_\beta$  estimates, we can plot the average estimated probability of making a field goal from any distance. The logistic regression estimate seems reasonable and represents the increasing difficulty of longer range kicks. Additionally, we can see how the model has changed from the initial guess of the prior, shown as the dashed line below. It seems that the first guess didn't give enough weight to closer kicks, but we were operating on very limited data.

```
# Average coefficients
avg.alpha <- summary(fit.samples)$statistics[19]
avg.beta <- summary(fit.samples)$statistics[20]
paste("Avg. Kicker alpha:", avg.alpha); paste("Avg. Kicker beta", avg.beta)
```

```
## [1] "Avg. Kicker alpha: 4.35349525225737"
```

```
## [1] "Avg. Kicker beta -0.0673565392596511"
```

```
# Find logistic curve for average
x <- seq(0,100,by=0.1)
kavg <- 1 / (1 + exp(-(avg.alpha + avg.beta * x)))
# Prior
kprior <- 1 / (1 + exp(-(3-5.125/100*x)))
# Plot
plot(x, kavg, main="Probability of Made FG - League Avg.", xlab="Distance",
     ylab="Make Prob. theta", ylim=c(0,1), col="blue", type="l")
lines(x, kprior, col="purple", lty=2)
legend(0, 0.4, legend=c("Posterior", "Prior"), lty=c(1,2),
      col=c("blue", "purple"))
```

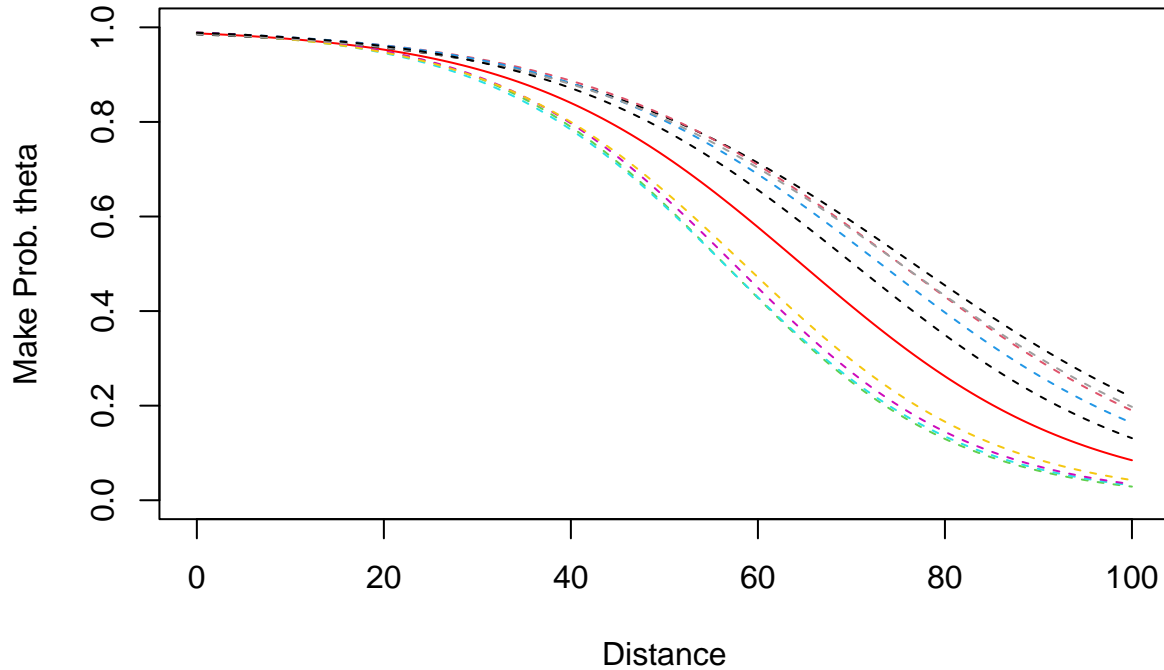


Further, we can plot the probability lines for all the kickers to see the variation between their abilities.

```
plot(x, kavg, main="Probability of Made FG - League Avg. Spread", xlab="Distance",
     ylab="Make Prob. theta", ylim=c(0,1), col="red", type="l")
for (i in 1:9){
  a <- summary(fit.samples)$statistics[i]
  b <- summary(fit.samples)$statistics[i+9]

  k <- 1 / (1 + exp(-(a + b * x)))
  lines(x, k, col=i, lty=2)
}
```

## Probability of Made FG – League Avg. Spread



Before diving into individual players, we can see there is a group of 4 below average kickers and 5 above-average kickers, which are spread out around the prior. This could be done for a much larger sample size, say all kickers in the league or free agency, or perhaps at a college level for the upcoming draft.

## Individual Kicker Evaluations

To explore some of the players a little more in depth, we proposed an answer for the following scenarios:

- Kicker A has a league-average contract salary and feels he should be paid more based on his performance.
- In a game-time situation, player B's coach only wants to attempt a 40-yard field goal if the kicker's probability is 90%.

We can use the results from before to explore both scenarios.

### Kicker A

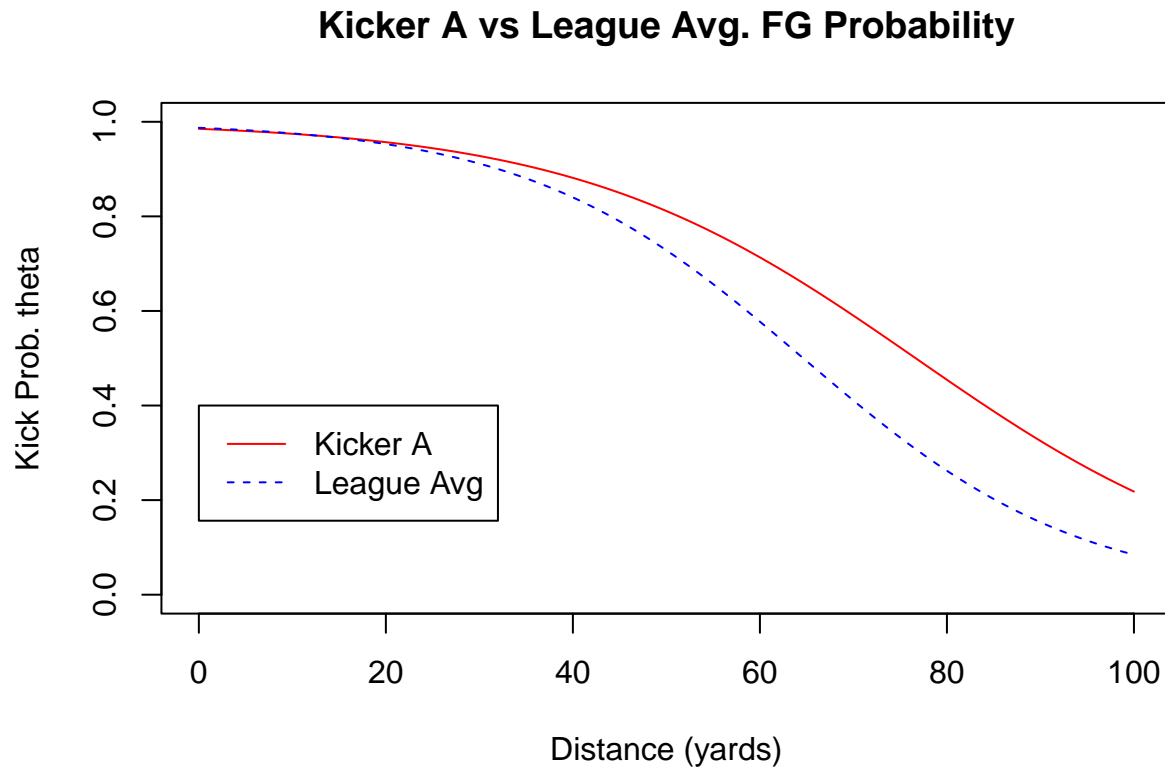
To see if kicker A is an above average kicker in the league and deserves a better contract, we can graph the logistic regression with his values versus the league average, based on the sampled  $\mu_\alpha, \mu_\beta$  values. Below is the transformed plot of the probability of a successful kick  $\theta$  as a function of distance.

```
## [1] "Kicker A alpha: 4.19475486533102"
```

```
## [1] "Kicker A beta: -0.0547181571788221"
```

Therefore, we can model the average probability of a made FG as

$$\text{logit}(\theta) = 5.312 - 0.087 * \text{Distance}$$



According to the model, it seems that kicker A has an above average probability of making a field goal at every distance. According to the coefficients, kicker A has much lower  $\beta$  value, meaning that he has higher probability of making field goals from more distance.

This can easily be seen in the plot, as the red line clearly lifts above the average line, seeing a noticeable gap at even as few as 30 yards out. Even at 100 yards, the model gives this kicker some probability of making it, even though that probably isn't realistic.

Either way, this model supports that kicker A is due for a pay raise, since he is clearly performing above the average kicker, at least from our observed data.

## Kicker B

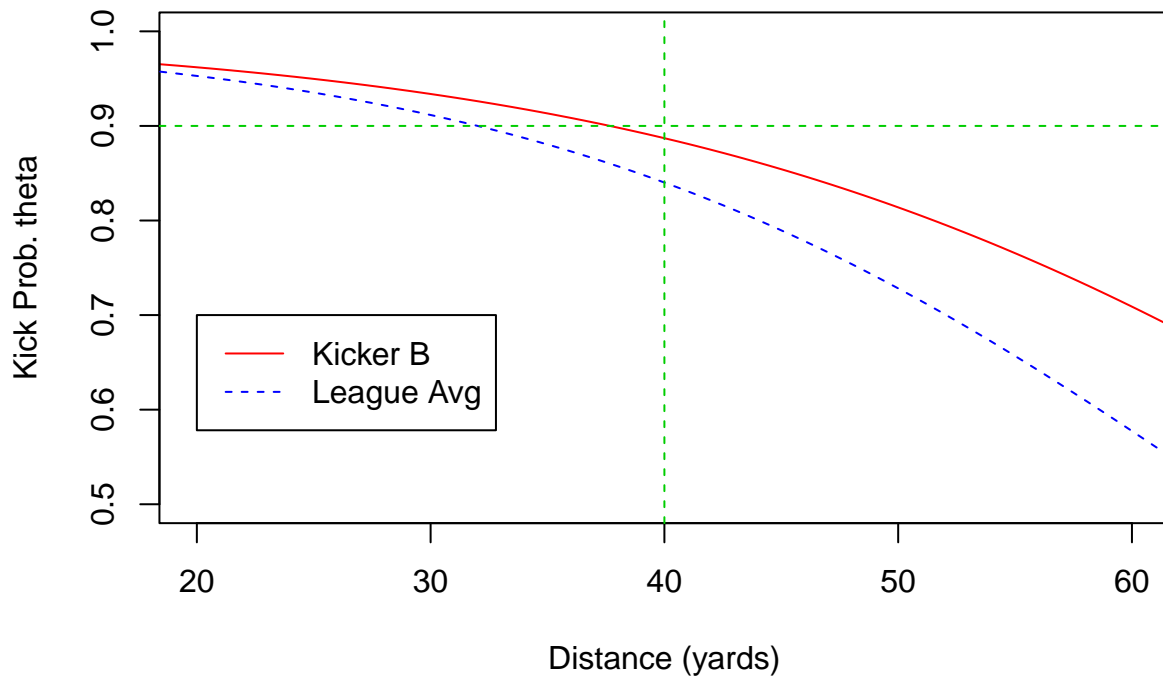
In a game situation, is kicker B dependent enough to try a field goal from 40 yards out? We can again model his probability of a made field goal like kicker A, and look at what his chances are at 40 yards, and we'll try his leg if  $\theta \geq .90$

```
## [1] "Kicker B alpha: 4.40227532596066"
```

```
## [1] "Kicker B beta: -0.0585359393990509"
```



### Kicker B vs League Avg. FG Probability



```
## [1] "Calculated probability of making 40-yd FG: 0.88704"
```

From the model, the probability of making a 40 yard field goal for kicker B is just under 90%, depending on how we want to round. It is worth noting that kicker B, like kicker A, is pretty well above average, and not many kickers would be given this high of an estimated probability in this situation.

While the model shows some possible deviation, with  $\alpha$  and  $\beta$  having sample standard deviations of 1.067 and 0.027, we can be pretty sure that the true probability is around 90%, since the estimate almost nearly passes right through that point.

While a healthy amount of skepticism is fair, expecting 90% accuracy from 40 yards is a stretch for most average NFL kickers. It appears kicker B has an above average kicking ability, and the model shows more confidence in him to make a 40-yarder. Since the estimate is almost exactly 90%, my recommendation would be to attempt the field goal since we have an exceptional kicker on hand.