Hierarchische Bayesianische Datenanalyse

(continuation of "Probabilistische Modellierung")

Building a prior model

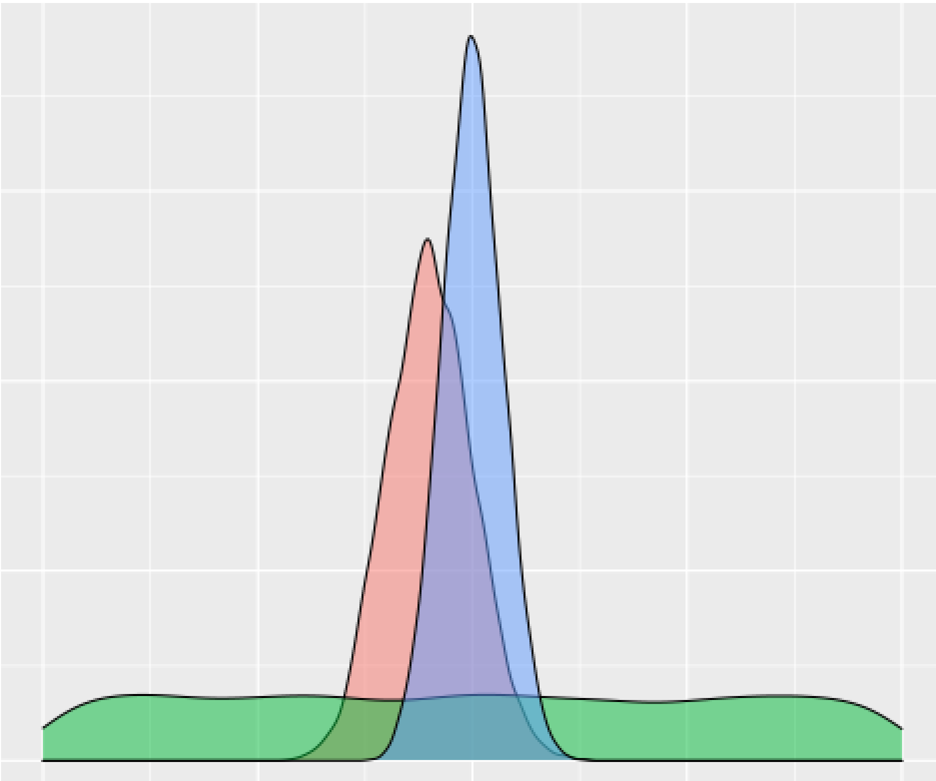
Beispiel: in einer wahl gibt es am anfang die schätzung, das wahrscheinlich 45%-55% der Stimmen an dich gehen.

Das bedeutet, dass der prior um 45/55 herum sein muss. Die art von Verteilung ist dabei nicht erzwungen, als beispiel kann man die beta verteilung verwenden. oder in R Studio:

```
prior_A <- rbeta(n=10000,45,55)
```

n = anzahl and "draws", gibt dir ein dataframe der größe 10000 mit der beta verteilung

mit unterschiedlichen parametern bekommt man (natürlich) unterschiedliche priors

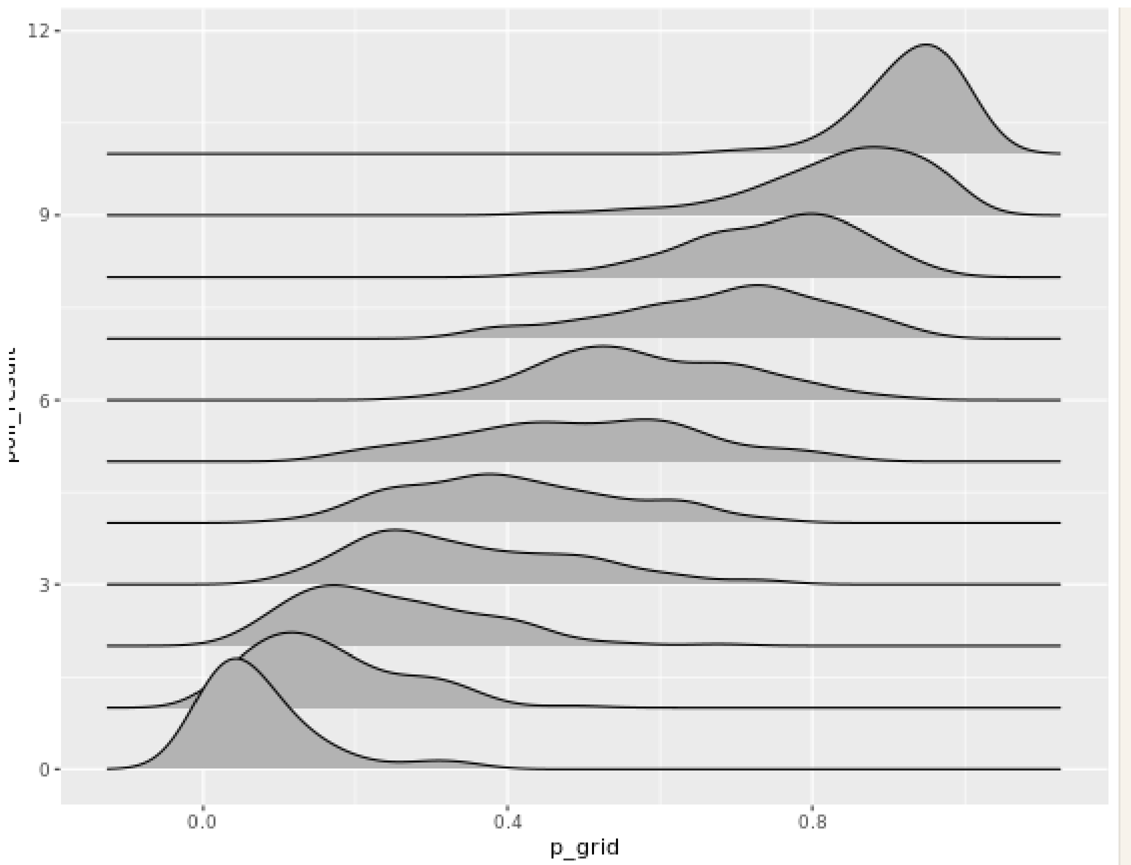


Likelyhood funktion

In dem Beispiel der Wahl ist das passendstes Modell die Binomial verteilung

(Wähler sind unabhängig, und es gibt nur 2 erreignise, gewählt oder nicht gewählt)

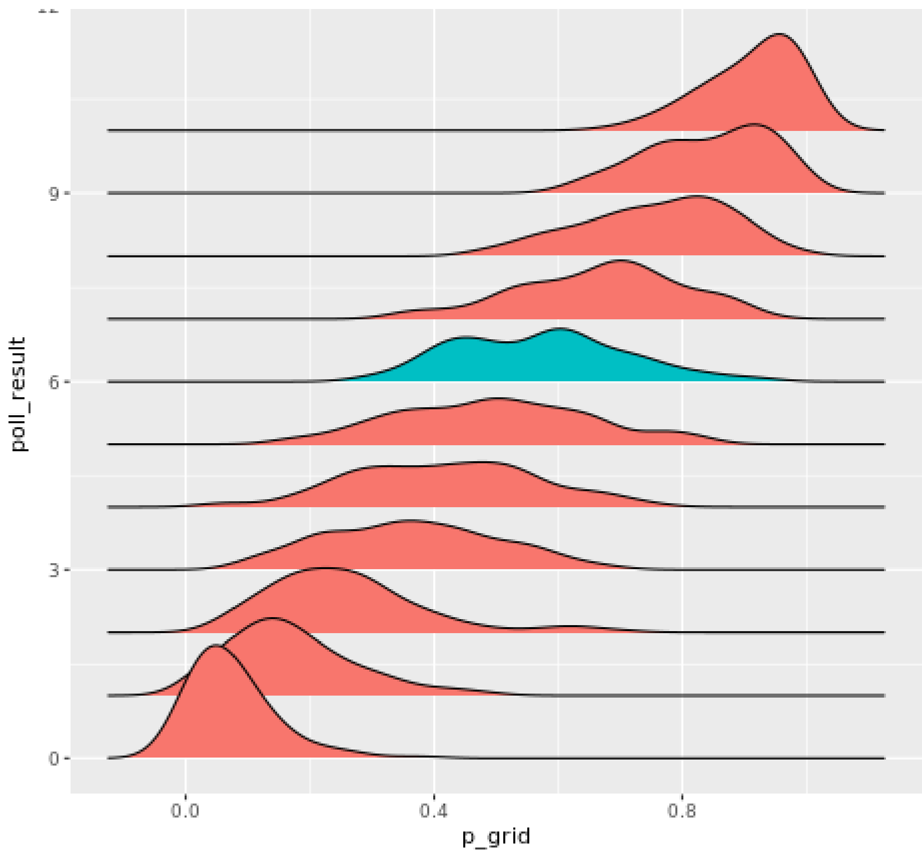
```r
# Define a vector of 1000 p values
p_grid <- seq(from = 0, to = 1, length.out = 1000)

# Simulate 1 poll result for each p in p_grid
poll_result <- rbinom(n = 1000, size = 10, prob = p_grid)

# Create likelihood_sim data frame
likelihood_sim <- data.frame(p_grid, poll_result)

# Density plots of p_grid grouped by poll_result
ggplot(likelihood_sim, aes(x = p_grid, y = poll_result, group = poll_result)) +
    geom_density_ridges()
```



je nachdem wie viele tatsächlich gewählt haben, gibt es unterschiedliche binomialverteilungen, also unterschiedliche likelyhood funktionen

(Wenn zb 6 gewählt haben, ist es die binomialverteilung in poll_result = 6)

```r
ggplot(likelihood_sim, aes(x = p_grid, y = poll_result, group = poll_result,
fill = poll_result == 6)) +
    geom_density_ridges()
```



Das ist dann die likelyhood funktion, was P ist, wenn 6 leute gewählt haben

Posterior in RJAGS

Wir wissen, dass posterior ~ prior x likelyhood ist, jedoch funktioniert diese formel tatsächlich nur selten.

```
# DEFINE the model
vote_model <- "model{
    # Likelihood model for X
    X ~ dbin(p, n)
    # Prior model for p
    p ~ dbeta(a, b)
}"
# COMPILE the model
vote_jags <- jags.model(textConnection(vote_model),
    data = list(a = 45, b = 55, X = 6, n = 10),
    inits = list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 100))
# SIMULATE the posterior
vote_sim <- coda.samples(model = vote_jags,
    variable.names = c("p"),
    n.iter = 10000)
```

Stadessen können wir RJAGS (eine kombination von R und JAGS) verwenden, um den Posterior zu simulieren.

Zuallererst macht man ein "vote model", welches sagt, welche modelle die prior und die likelyhood verwenden dann fügen wir in vote_jags die parameter hinzu (a 45, n 55, da wir im prior eine rund 50% chance erwarten, X = 6, da 6 leute richtig gewählt haben, und n = 10, da wir 10 Wähler hatten)

Dann können wir den posterior mit coda.samples simulieren (und das 10000 mal) und können die simulationen zusammenfassen in vote_sim

Normal normal model

Beispiel:

Man möchte wissen, wie schlafenzug die reaktionszeit beintrechtigt

Wir schätzen, dass die Veränderung der reaktionszeit Yi Normalverteilt ist, mit der Durchschnittlichen veränderung der Reaktionszeit als mittelwert, und s als Standardabweichung

Die Standardabweichung s muss größer als 0 sein, und mit normalen schlaf ist die SD ~30 (25) ms, trozdem schätzen wir s also irgendetwas zwischen 0 und 200 ms

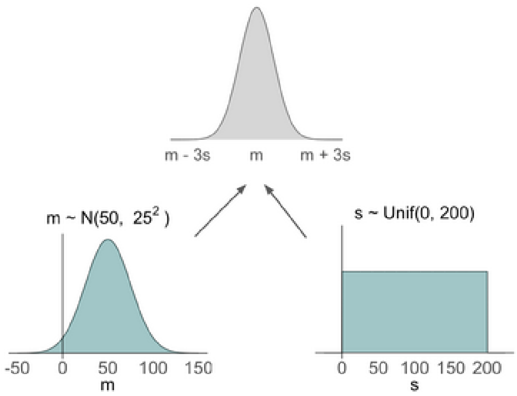
Auch schätzen wir, dass die mittlere Reaktionszeitveränderung m auch normal verteilt ist, mit 50 ms als der Mittelwert, und 25 als Standardabweichung.

Likelihood:

$$Y_i \sim N(m, s^2)$$

Priors:

$$m \sim N(50, 25^2)$$
$$s \sim \text{Unif}(0, 200)$$




```r
# Take 10000 samples from the m prior
prior_m <- rnorm(n = 10000, mean = 50, sd = 25)

# Take 10000 samples from the s prior
prior_s <- runif(n = 10000, min = 0, max = 200)

# Store samples in a data frame
samples <- data.frame(prior_m, prior_s)

# Density plots of the prior_m & prior_s samples
ggplot(samples, aes(x = prior_m)) +
    geom_density()
ggplot(samples, aes(x = prior_s)) +
    geom_density()
```