

**If you're a Bayesian you can do anything that
God forbids (- Willem Heiser)**

Joachim Vandekerckhove

Unorthodox things you can do if you're a Bayesian

- Accumulate evidence for the absence of an effect

Unorthodox things you can do if you're a Bayesian

- Accumulate evidence for the absence of an effect
- Draw valid conclusions with almost no data

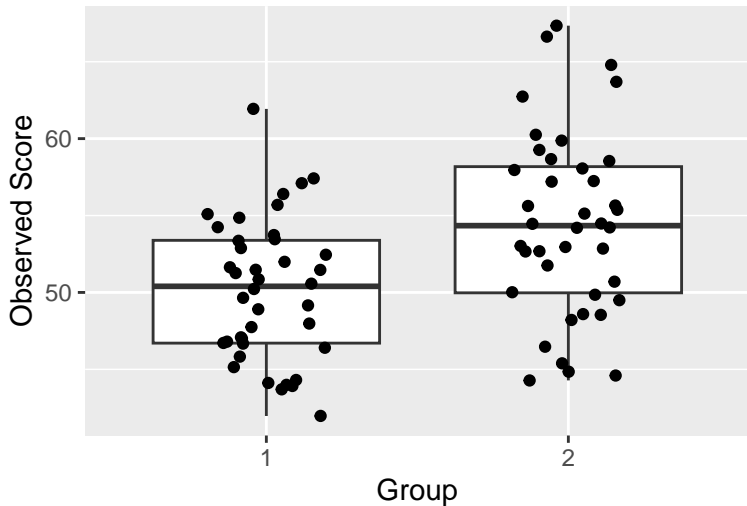
Unorthodox things you can do if you're a Bayesian

- Accumulate evidence for the absence of an effect
- Draw valid conclusions with almost no data
- Say things about means without knowing much about the basic units

Means without knowing much about the basic units

```
N <- 80
group <- rep(1:2, each = N/2)
true_score <-
  rnorm(N,
        mean = rep(c(50, 55), each = N/2),
        sd    = 2)  # True scores for each participant
observed_score <- true_score +
  rnorm(N,
        mean = 0,
        sd    = 5)  # Noisy observations
data <- data.frame(participant = 1:N,
                   group = factor(group),
                   true_score, observed_score)
```

Observed Scores by Group



```

model_string <- "
model {
  for (i in 1:N) {
    # Likelihood
    observed_score[i] ~ dnorm(true_score[i], tau)
    # Hierarchical model
    true_score[i]      ~ dnorm(mu[group[i]], tau_true)
  }
  diff_mu ~ dnorm(0, 0.001)
  mu[1]    ~ dunif(0, 100)
  mu[2] <- mu[1] + diff_mu
  tau      ~ dgamma(0.1, 0.1)
  tau_true ~ dgamma(0.1, 0.1)
}
"

```

```
data_list <- list(  
  N = N,  
  observed_score = data$observed_score,  
  group = as.numeric(data$group)  
)  
  
# Initial values  
inits <- function() {  
  list(  
    tau          = rgamma(1, 0.1, 0.1),  
    tau_true     = rgamma(1, 0.1, 0.1),  
    true_score   = runif(N, 0, 100)  
  )  
}
```



```
jags_model <- jags.model(textConnection(model_string),  
                          data      = data_list,  
                          inits     = inits,  
                          n.chains = 3 ,  
                          n.adapt  = 1000 )
```

```
## Compiling model graph  
##   Resolving undeclared variables  
##   Allocating nodes  
## Graph information:  
##   Observed stochastic nodes: 80  
##   Unobserved stochastic nodes: 84  
##   Total graph size: 250  
##  
## Initializing model
```

```

update(jags_model, n.iter = 1000)
samples <- coda.samples(jags_model,
                        variable.names = c("diff_mu",
                                           "true_score"),
                        n.iter = 5000)

```

```

# Summarize the results
summary_diff_mu <- summary(samples)$
  statistics["diff_mu", ]
print(summary_diff_mu)

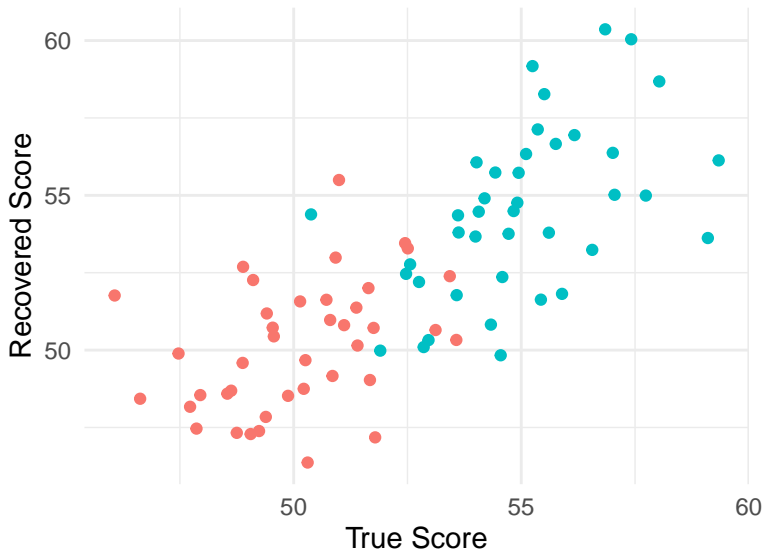
```

##	Mean	SD	Naive SE	Time-series SE
##	4.3570	1.2028	0.0098	0.0692

```
# Extract true scores from the posterior samples
true_scores_posterior <- as.data.frame(as.matrix(samples)) %>%
  select(starts_with("true_score")) %>%
  apply(2, mean)

# Add the recovered true scores to the data frame
data$recovered_true_score <- true_scores_posterior

# Scatter plot of true scores vs recovered true scores
p2 <- ggplot(data, aes(x = true_score, y = recovered_true_score,
                       color = group)) +
  geom_point() +
  labs(x = "True Score", y = "Recovered Score") +
  theme_minimal() +
  theme(legend.position = "none")
```



Hierarchical recovery beats individual recovery

The difference between groups is well recovered as 4.36 ± 1.2 , even though the true scores within each group are poorly recovered.

