Chapter 12 - Methods for Bayes Null Hypothesis Testing

Sunday, April 3, 2022 4:40 PM

Resource of ROPE:

https://cran.r-project.org/web/packages/bayestestR/vignettes/region_of_practical_equivalence.html

ROPE: Region of practical significance

Rejected ROPE:

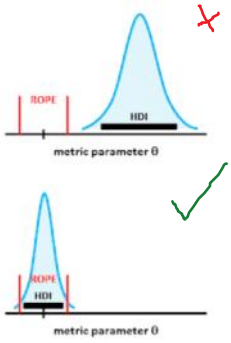
- Parameter is not credible if the ROPE does not lie within the 95% HDI
- A parameter value is declared to be not credible, or rejected, if its entire ROPE lies outside the 95% highest density interval (HDI) of the posterior distribution of that parameter.

Accepted ROPE:

- if the rope lies within the HDI'
- A parameter value is declared to be accepted for practical purposes if that value's ROPE completely contains the 95% HDI of the posterior of that parameter.
- With this decision rule, a null value of a parameter can be accepted only when there is sufficient precision in the estimate of the parameter.

Decision by posterior HDI and ROPE

- Consider a landmark parameter value. Values that are equivalent to that landmark for practical purposes define the ROPE around that value.
  - ROPE is specified according to utilities and/or predictions of competing theories. Routinely done in frequentist *equivalence testing* and clinical *non-inferiority testing*.
- The ROPEd parameter value is rejected if its ROPE excludes the 95% HDI.
  - Rejects the ROPEd value, not entire ROPEd interval.
- The ROPEd parameter value is accepted for practical purposes if its ROPE includes the 95% HDI.
  - Accepts the ROPEd value, not entire ROPEd interval.
- If ROPE and 95% HDI overlap, neither accept nor reject the ROPEd parameter value.



How does Negative and Positive Correlation affect the Testing?

Positive Correlation: NARROW interval. Negative Correlation: WIDE interval

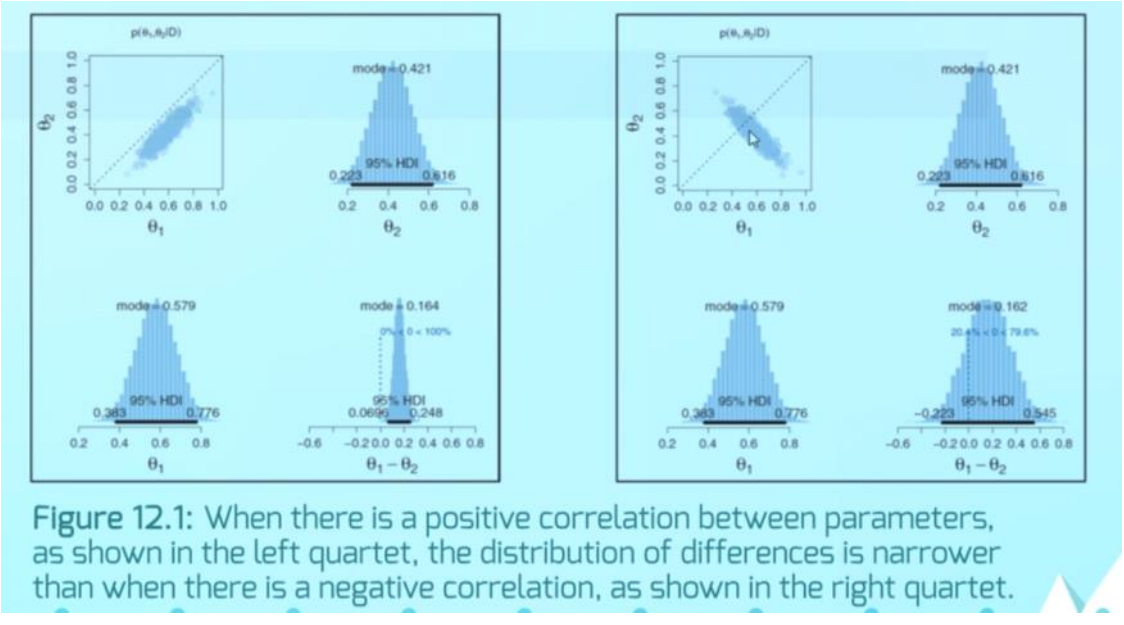
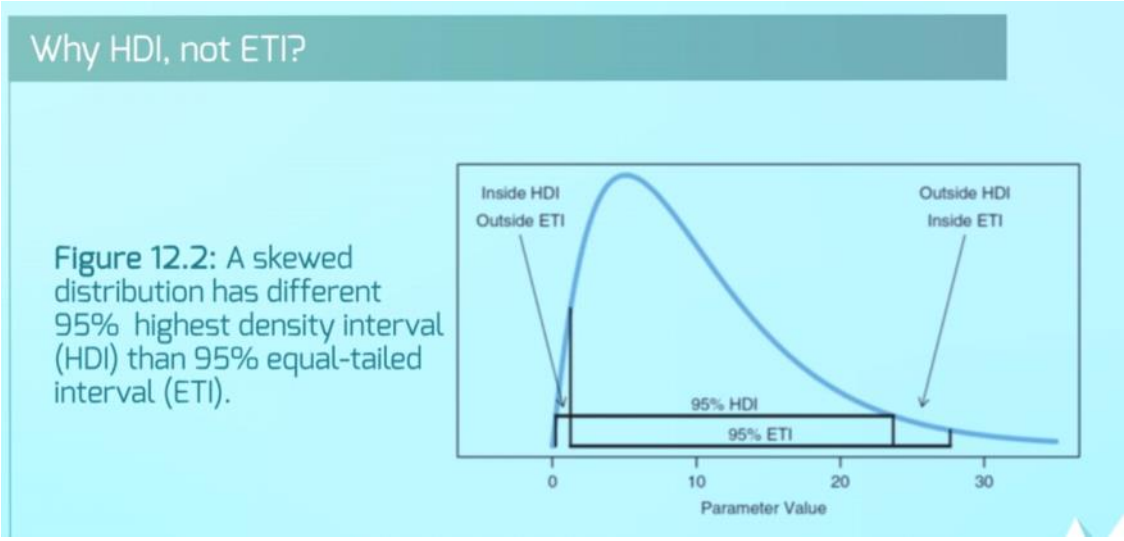


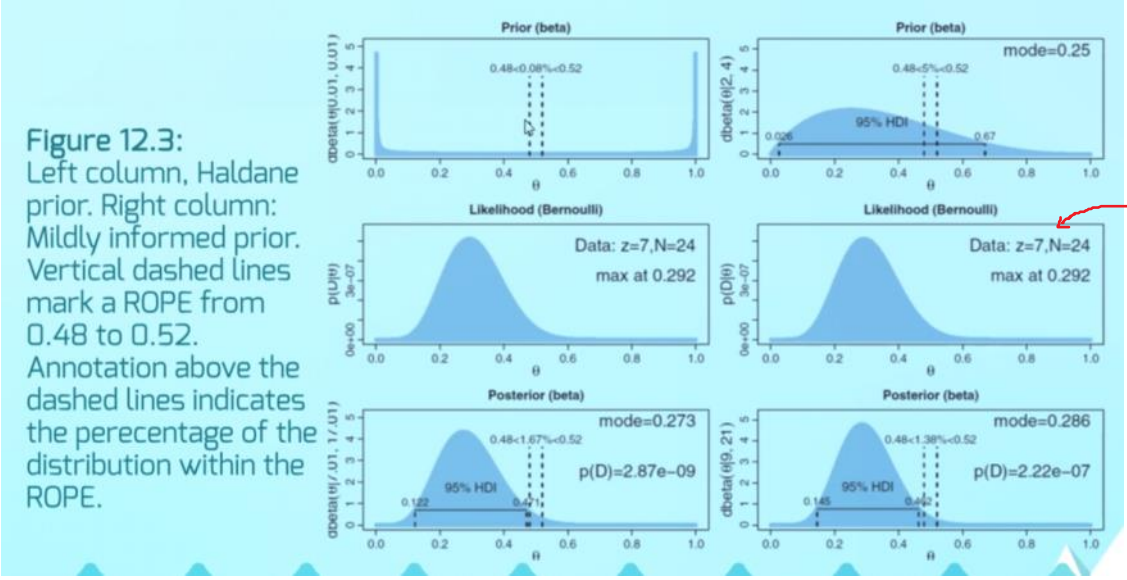
Figure 12.1: When there is a positive correlation between parameters, as shown in the left quartet, the distribution of differences is narrower than when there is a negative correlation, as shown in the right quartet.

Why choose HDI for this comparison and not ETI?

HDI is a narrower estimate and will allow for better comparison

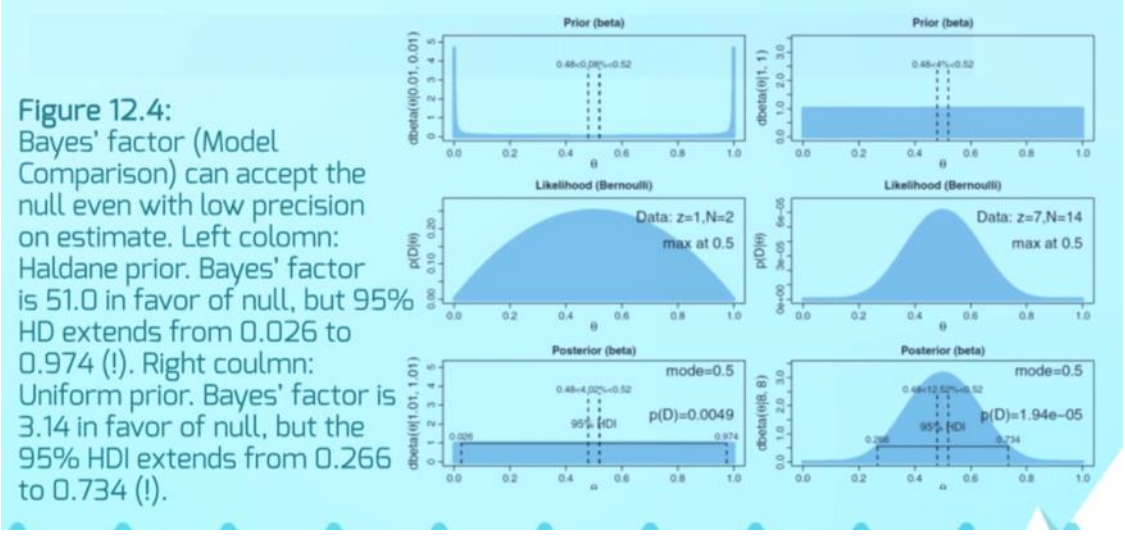


See the effects of the ROPE



7 out of 24 accepted

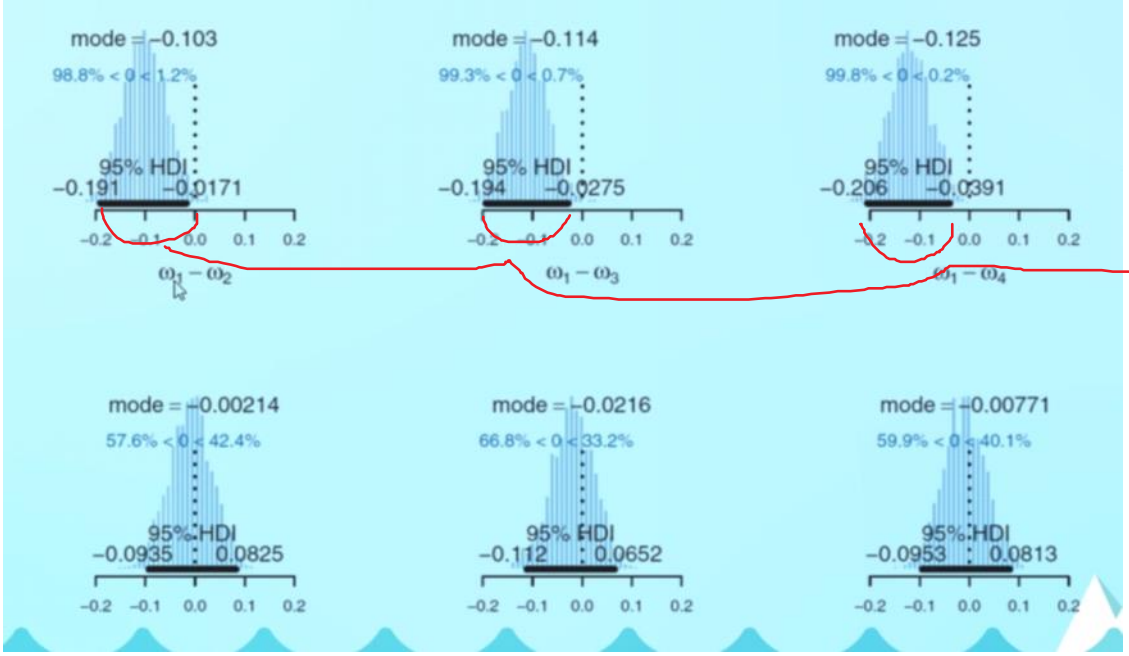
Bayes factors can accept null with poor precision (bot good)



Two Approaches to Estimation about if groups are equal:

1. Estimation Approach

- Measure an effect of difference in two groups
- Good approach and probably the best way to go



2. Model Selection Approach

- JAGS uses this approach
- Shows if the two groups are different from each other

JAGS Model - Using Psuedo Priors

The data structure has one row per subject, with the number of trials (words) for subject `s` denoted `nTrlOfSubj[s]`, the number correctly recalled for subject `s` denoted `nCorrOfSubj[s]`, and the condition of subject `s` denoted `CondOfSubj[s]`. The model specification begins with saying that each subject has an individual ability `theta[s]` from a condition-specific beta distribution:

```
model {
  for ( s in 1:nSubj ) {
    nCorrOfSubj[s] ~ dbin( theta[s] , nTrlOfSubj[s] )
    theta[s] ~ dbeta( aBeta[CondOfSubj[s]] , bBeta[CondOfSubj[s]] )
  }
```

The shape parameters of the beta distribution are then re-written in terms of the mode and concentration. Model 1 uses condition-specific `omega[j]`, while Model 2 uses the same `omega0` for all conditions. The JAGS function `equals(mdlIdx,...)` is used to select the appropriate model for index `mdlIdx`:

```
  for ( j in 1:nCond ) {
    # Use omega[j] for model index 1, omega0 for model index 2:
    aBeta[j] <-       ( equals(mdlIdx,1)*omega[j]
                      + equals(mdlIdx,2)*omega0  )   * (kappa[j]-2)+1
    bBeta[j] <- ( 1 - ( equals(mdlIdx,1)*omega[j]
                      + equals(mdlIdx,2)*omega0  ) ) * (kappa[j]-2)+1
    omega[j] ~ dbeta( a[j,mdlIdx] , b[j,mdlIdx] )
  }
  omega0 ~ dbeta( a0[mdlIdx] , b0[mdlIdx] )
```

The priors on the concentration parameters are then specified:

```
for ( j in 1:nCond ) {
  kappa[j] <- kappaMinusTwo[j] + 2
  kappaMinusTwo[j] ~ dgamma( 2.618 , 0.0809 ) # mode 20 , sd 20
}
```

How the Pseudo Priors were Chosen

```
# Constants for prior and pseudoprior:
aP <- 1
bP <- 1
# a0[model] and b0[model]
a0[1] <- 0.48*500       # pseudo
b0[1] <- (1-0.48)*500   # pseudo
a0[2] <- aP             # true
b0[2] <- bP             # true
# a[condition,model] and b[condition,model]
a[1,1] <- aP            # true
a[2,1] <- aP            # true
a[3,1] <- aP            # true
a[4,1] <- aP            # true
b[1,1] <- bP            # true
b[2,1] <- bP            # true
b[3,1] <- bP            # true
b[4,1] <- bP            # true
a[1,2] <- 0.40*125      # pseudo
a[2,2] <- 0.50*125      # pseudo
a[3,2] <- 0.51*125      # pseudo
a[4,2] <- 0.52*125      # pseudo
b[1,2] <- (1-0.40)*125  # pseudo
b[2,2] <- (1-0.50)*125  # pseudo
b[3,2] <- (1-0.51)*125  # pseudo
b[4,2] <- (1-0.52)*125  # pseudo
# Prior on model index:
mdlIdx ~ dcat( modelProb[] )
modelProb[1] <- 0.5
modelProb[2] <- 0.5
}
```