BDA HM8

Piero Birello

February 2025

13.2

(A) The code used is the following:

```
source("minNforHDIpower.R")

sampSize = minNforHDIpower( genPriorMode=0.8, genPriorN=2000,
                            HDImaxwid=0.20, nullVal=NULL, ROPE=NULL,
                            desiredPower=0.8,
                            audPriorMode=0.5, audPriorN=2,
                            HDImass=0.95, initSampSize=50, verbose=TRUE )
```

We obtain that for sample size = 67, power = 0.8239177. The The required N has decreased with respect to the case $\kappa = 10$.

(B) One may want to pursue the goal of precision even though the data generating distribution is already sharp in the case when the audience prior does not agree the formulated hypothesis. This may be the case if we have good evidence or belief about an hypothesis that goes against the most supported one in the field of study.

(C) The code used is the following:

```
sampSize = minNforHDIpower( genPriorMode=0.75, genPriorN=2,
                            HDImaxwid=NULL, nullVal=0.5, ROPE=c (0.48,0.52),
                            desiredPower=0.8,
                            audPriorMode=0.5, audPriorN=2,
                            HDImass=0.95, initSampSize=5, verbose=TRUE )
```

We obtain that for sample size = 134, power = 0.8. The required N has notably increased with respect to the case $\kappa = 2000$.

(D) When the prior distribution for the data-generating parameter is a beta distribution with $\mu = 0.8$ and $\kappa = 2$, the proportion of data generating biases that are higher than the null value $\theta = 0.5$ is 50%. Actually, in terms of shape parameters, this is a $\beta(1, 1)$, which is flat in $[0, 1]$. If our goal is to see the HDI to fall entirely above the null value, that is an unfeasible goal. The best Bayesian inference can do is correctly approximating the data generating distribution. Increasing the sample size, will let us converge towards a power of 0.5.

13.3

(A) Changing the number of simulated datasets to 50, we get similar power estimates to the table shown in the book. Actually, the data-generating distribution has not changed nor have any priors. However, the bounds we obtain are wider, meaning we can evaluate power with less certainty. Here is the obtained output:

```
[1] "omegaAboveROPE: Est.Power=1; Low Bound=0.943; High Bound=1"
[1] "omegaNarrowHDI: Est.Power=1; Low Bound=0.943; High Bound=1"
[1] "thetasAboveROPE: Est.Power=1; Low Bound=0.943; High Bound=1"
[1] "thetasNarrowHDI: Est.Power=0.32; Low Bound=0.203; High Bound=0.454"
```

(B) The chosen mode and standard deviation correspond to the ones of the posterior estimate of ω obtained from data. Such data consists in records of 28 practitioners making 10 trials each, from which the choice of the remaining parameters.

(C) Results after running the experiment are:

```
[1] "omegaAboveROPE: Est.Power=0; Low Bound=0; High Bound=0.133"
[1] "omegaNarrowHDI: Est.Power=1; Low Bound=0.867; High Bound=1"
[1] "thetasAboveROPE: Est.Power=0; Low Bound=0; High Bound=0.133"
[1] "thetasNarrowHDI: Est.Power=0.4; Low Bound=0.213; High Bound=0.61"
```

While the power for obtaining a narrow HDI for ω is high, the power for a narrow HDI is low inthe case of θ. Indeed, the first distribution is narrow by construction while the second one is not, since it accounts for variations among subjects.

(D) Using real data instead of idealized ones, we obtain:

```
[1] "omegaAboveROPE: Est.Power=0; Low Bound=0; High Bound=0.133"
[1] "omegaNarrowHDI: Est.Power=0.7; Low Bound=0.491; High Bound=0.864"
[1] "thetasAboveROPE: Est.Power=0; Low Bound=0; High Bound=0.133"
[1] "thetasNarrowHDI: Est.Power=0.15; Low Bound=0.041; High Bound=0.34"
```

Power estimates are lower than the last case, since here we are setting the actual numbers of participants and trials (28,10) which is lower than the idealized ones we used (40,100).

13.4

```
source("DBDA2E-utilities.R")

fig_path = "Figures/NHTStopping/"
openGraph(width=6,height=6)

# Set up the plotting area for 5 panels
par(mfrow=c(5,1), mar=c(3,4,2,2))

# Set parameters
pHeads <- 0.5  # Underlying probability
max_N <- 1000   # Maximum number of trials
theta_null <- 0.5  # Null hypothesis value
a_alt <- 1  # Prior alpha for alternative
b_alt <- 1  # Prior beta for alternative
ROPE_semiwidth = 0.05

# Generate the flip sequence for all trials
set.seed(15)  # For reproducibility
flipSequence <- sample(x=c(0,1), prob=c(1-pHeads, pHeads), size=max_N, replace=TRUE)

# Initialize vectors to store results for each N
n_values <- 1:max_N
runProp_values <- numeric(max_N)
pValue_values <- numeric(max_N)
logBF_values <- numeric(max_N)
hdiWidth_values <- numeric(max_N)
hdi_lower <- numeric(max_N)
```

```
hdi_upper <- numeric(max_N)

# Initialize vectors to store decisions: -1=reject, 0=don't know,1=accept
p_decision <- integer(max_N)
BF_decision <- integer(max_N)
HDI_decision <- integer(max_N)

# Calculate metrics for each value of N
for (i in 1:max_N) {
  # Current N and data
  N <- i
  z <- sum(flipSequence[1:N])

  # Running proportion
  runProp_values[i] <- z / N

  # P-value
  pValue_values[i] <- binom.test(x=z, n=N, p=theta_null, alternative="two.sided")$p.value
  p_decision[i] <- -(pValue_values[i]<0.05)

  # Bayes Factor
  p_D_given_alt <- beta(z+a_alt, N-z+b_alt)/beta(a_alt,b_alt)
  p_D_given_null <- theta_null^z * (1-theta_null)^(N-z)
  logBF_values[i] <- log(p_D_given_alt)-log(p_D_given_null)
  if (logBF_values[i]>1){BF_decision[i] <- -1}
  else if (logBF_values[i]<(-1)){BF_decision[i] <- 1}
  else {BF_decision[i] <- 0}

  # HDI
  EstHDI <- HDIofICDF(qbeta, shape1=z+a_alt, shape2=N-z+b_alt)
  hdi_lower[i] <- EstHDI[1]
  hdi_upper[i] <- EstHDI[2]
  hdiWidth_values[i] <- EstHDI[2] - EstHDI[1]
  rope_max = theta_null+ROPE_semiwidth
  rope_min = theta_null-ROPE_semiwidth
  if (hdi_lower[i] > rope_max || hdi_upper[i] < rope_min) {HDI_decision[i] <- (-1)}
  else if (hdi_lower[i] >= rope_min && hdi_upper[i] <= rope_max) {HDI_decision[i] <- 1}
  else {HDI_decision[i] <- 0}
}

# Convert decisions to colors
colors <- c("red", "grey", "blue")
names(colors) <- c("-1", "0", "1")
p_color <- colors[as.character(p_decision)]
BF_color <- colors[as.character(BF_decision)]
HDI_color <- colors[as.character(HDI_decision)]
print(HDI_decision)

# Panel 1: Running Proportion
plot(n_values, runProp_values, type="o", col="black",
     xlim=c(1,max_N), ylim=c(0.0,1.0), cex.axis=1.2,
     xlab="", ylab="Proportion")
abline(h=pHeads, lty="dotted")
# Display info
flipLetters <- paste(c("T","H")[flipSequence[1:10]+1], collapse="")
displayString <- paste0("Flip Sequence = ", flipLetters, "...")
#text(max_N, 0.9, displayString, adj=c(1,0.5), cex=1.0)
text(max_N, 0.8, paste("End Proportion =", round(runProp_values[max_N], 3)), adj=c(1,0.5), cex
    =1.0)

# Panel 2: P-values
plot(n_values, pValue_values, type="o", col=p_color,
```

```
     xlim=c(0,max_N), ylim=c(0,1), cex.axis=1.2,
     xlab="", ylab="p-value")
abline(h=0.05, lty="dashed", col="black")
text(max_N, 0.9, paste("Final p-value =", round(pValue_values[max_N], 4)), adj=c(1,0.5), cex
    =1.0)

# Panel 3: Log Bayes Factor
plot(n_values, logBF_values, type="o", col=BF_color,
     xlim=c(0,max_N), cex.axis=1.2,
     xlab="", ylab="log(BF)")
abline(h=1, lty="dashed", col="black")
abline(h=-1, lty="dashed", col="black")
text(max_N, min(logBF_values, na.rm=TRUE) + 0.9*(max(logBF_values, na.rm=TRUE) - min(
    logBF_values, na.rm=TRUE)),
     paste("Final log(BF) =", round(logBF_values[max_N], 2)), adj=c(1,0.5), cex=1.0)

# Panel 4: HDI Bounds
#fill_color <- adjustcolor(HDI_color, alpha.f = 0.3)
plot(n_values, hdi_lower, type="o", col="white",
     xlim=c(0,max_N), ylim=c(0,1), cex.axis=1.2,
     xlab="N", ylab="HDI Bounds")
# Fill the area between the bounds
#polygon(c(n_values, rev(n_values)),
#        c(hdi_lower, rev(hdi_upper)),
#        col=fill_color, border=NA)
# Number of steps
n_steps <- length(n_values) - 1
# Loop through each segment and fill with a varying color
for (i in 1:n_steps) {
  # Define color for this step
  step_color <- HDI_color[i]
  # Draw a small polygon for each segment
  polygon(c(n_values[i], n_values[i+1], n_values[i+1], n_values[i]),
          c(hdi_lower[i], hdi_lower[i+1], hdi_upper[i+1], hdi_upper[i]),
          col = step_color, border = NA)
}
abline(h=pHeads-ROPE_semiwidth, lty="dashed", col="black")
abline(h=pHeads+ROPE_semiwidth, lty="dashed", col="black")
text(max_N, 0.9, paste("Final HDI: [", round(hdi_lower[max_N], 3), ",", round(hdi_upper[max_N
    ], 3), "]"),
     adj=c(1,0.55), cex=1.0)

# Panel 5: HDI Width
plot(n_values, hdiWidth_values, type="o", col="black",
     xlim=c(0,max_N), ylim=c(0, max(hdiWidth_values, na.rm=TRUE) * 1.1), cex.axis=1.2,
     xlab="", ylab="HDI Width")
abline(h=0.1, lty="dashed", col="black")
text(max_N, max(hdiWidth_values, na.rm=TRUE) * 0.9,
     paste("Final HDI Width =", round(hdiWidth_values[max_N], 3)), adj=c(1,0.45), cex=1.0)

saveGraph(file=paste(fig_path,"NHTComparison"),type="pdf")
```

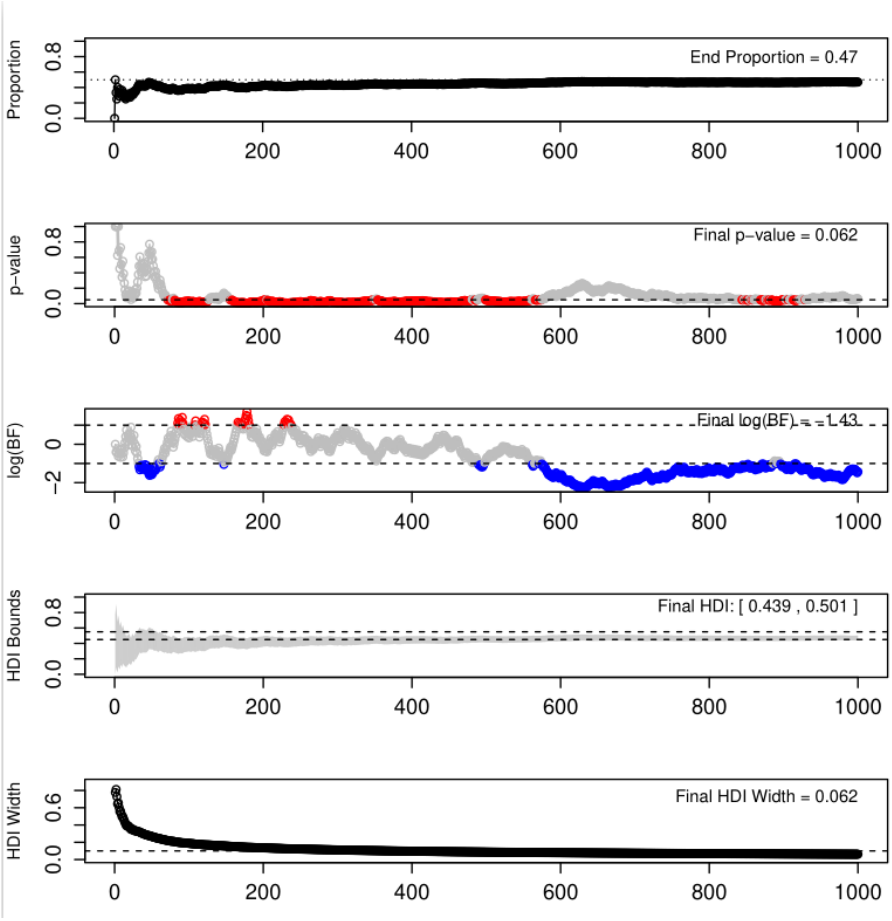

Figure 1: Running proportion of heads, p-value (assuming fixed sample size), log Bayes Factor, HDI of the posterior distribution for the bias, and HDI width. For the three central panels, points are red when the null hypothesis is rejected, blue when it is accepted and gray when no decision is made. We show a particularly 'unlucky' sequence. The decision rule based on the HDI is the most safe: no decision is taken, we would require more data. Both the p-value and Bayes Factor reject the -correct- null hypothesis at some point.