

```

# Simulation of Diversity in Variation Estimates

library(tidyverse) # for plot at end
set.seed(42) # reproducible

# GSI function from submission
CalcGiniSimpsonDivIndex<-function(array_of_counts){
  array_of_fractions = array_of_counts/sum(array_of_counts)
  squared_array_of_fractions = array_of_fractions^2
  sum_squared_array_of_fractions = sum(squared_array_of_fractions)
  Svalue = 1-sum_squared_array_of_fractions
  return(Svalue)
}

# Empty data frame
testingVar <- data.frame(SampleSize = numeric(),
                        MeasuredGSI = numeric(),
                        NormalisedGSI = numeric())

# Looking at some four samples (of different sizes) from a population
with fixed diversity characteristics
sample_sizes <- c(10, 20, 50, 100)
STs <- LETTERS[1:4] # real richness (assuming there are exactly 4
types)
frequencies <- c(0.5, 0.3, 0.1, 0.1) # real frequencies in population

# Main loop with 1000 replicates (hard-coded)
for (n in sample_sizes) {
  TheorMaxGSI <- 1-sum(rep(1/n,n)^2) # from submission
  CalcGSI <- vector(length = 1000) # prefilled vector (for speed)
  for (i in 1:1000) {
    simdat <- data.frame(table(sample(STs, size=n, replace=T,
prob=frequencies))) # similar to sub
    CalcGSI[i] <- CalcGiniSimpsonDivIndex(simdat$Freq)
  }
  NormGSI <- CalcGSI / TheorMaxGSI # vectorised
  testingVar <- rbind( testingVar, data.frame(SampleSize = rep(n,
1000),
                                              MeasuredGSI = CalcGSI,
                                              NormalisedGSI = NormGSI) )
}

# Plot results and save to file
testingVar |>
  pivot_longer(cols = MeasuredGSI:NormalisedGSI, names_to = "Measure",
values_to = "GSI") |>
  mutate(SampleSize = factor(SampleSize)) |>
  ggplot(aes(x = SampleSize, y = GSI)) +
  geom_boxplot() + geom_jitter(alpha = 0.2, width = 0.2) +
  geom_hline(yintercept = CalcGiniSimpsonDivIndex(frequencies),
            linetype="dashed", color = "red", linewidth = 1) +
  facet_grid(~ Measure)
ggsave("simulated.pdf")

```

