The aim of this app is to use Bayesian inference for estimating the posterior beliefs of the bias of a coin. Suppose there is a mint, but you are not sure if the coins it mints are totally fair. One way to find out the bias of the mint is to flip several of its coins and estimate the probability of them landing heads. A Frequentist approach would require making a large number of flips, rely on the Central Limit Theorem, and computing the probability of the data given the null model is true. Another approach is to use Bayesian inference. In Bayesian inference, on the other hand, one poses an initial guess as to the possible bias of the coin (called the prior belief). One then flips a certain amount of coins and computes the observed probability of a coin landing heads (called the likelihood). Finally, one updates the initial belief in one’s guess by weighing in the observed evidence (called the posterior belief). This simulation allows you to explore how modifying the prior belief the estimation of the posterior. That is, one can explore how the number of coin flips (or the strength of the evidence), the initial guess as to the possible bias, or the certainty about the initial guess, systematically affect the distribution of the posterior. The simulation uses a Monte Carlo Markov Chain method to calculate the posterior distribution. The prior is specified as a normal distribution with mean equal to the initial guess, and standard deviation equal to the inverse of the precision around that initial guess. The likelihood is computed as a Bernoulli distribution with equal to N flips, and probability given by every step in the MCMC random walk.

Instructions

Select how many times you want to flip the coin (30 coins by default). Select your prior guess of the bias (if you are not sure, you can leave the default option which is 50-50). Select how certain you are about your guess (the larger the value the more certain you are that your guess is right). Finally, select how many steps do you want to allow the algorithm to take exploring the posterior space (by default, the algorithm discards the first 10% of the steps as a burn-in period).

For more information about the employed algorithm, please go to my website containing a presentation, or consult the original source (Kirschke, 2010).