Bayesian inference

Complete two of the three exercises.

1. Coin flips

File *Coin_flips.R* contains code for Bayesian analysis of coin flips. The posterior distribution is a standardized product of prior and likelihood. We will try to understand this changing the number of coin flips used for priors and likelihoods.

- Equal number of coin flips for prior and likelihood:
  - Compare modes (maximum value) of prior, likelihood and posterior

  - Does the absolute number of coin flips matter?
- Change number of coin flips used as prior
  - no coin flips
  - half the the data
  - twice the data
- Change number of coin flips used as data (always keep the proportion of heads at 0.75*n)
  - only four coin flips
  - twice the number of prior flips
  - five times the number of prior flips
- divide your coin flips into two parts:
  - 20 flips then use the new posterior as prior and add an additional 20 coin flips
  - compare this to 40 coin flips

2. Markov chain Monte Carlo sampling

Write a sample testing the assertion that the porption of visits to a state will converge to proportion of population.

- Person wants to visit all states in the US repeatedly, number of visits depends on population size
- Start with a randomly chosen state
- Draw a new state at random
- Compare populations of the two states
- ratio $\frac{P_{prop}}{P_{old}}$
- if ratio > 1 move, else ratio is probability of moving

Repeat many times

Proportion of visits will converge to proportion of population

3. Compare Bayesian and classical linear regressions

Run linear regression a dataset (dependent variable and independent variable) of your choice. If you don't have interesting data, take some climate data available in the *analogue* or *palaeoSig* packages. Use the original data as independent variable, then add white noise *rnorm* to the original data and use the new variable as dependent variable.

Run a Bayesian linear regression using the *MCMCregress* regress function available in the *MCMCpack* r-package. Per default, *MCMCregress* uses uniform priors for a and b.

The function will return a number (per default 10000) of draws from the posterior of a, b and s^2.

Make predictions without accounting for the variance component (a+bx) for each of the posterior draws. Estimate uncertainties of this ensemble. Compare these uncertainties to *predict(my.lm,interval='confidence')*.

Make predictions accounting for the variance component (a+bx+e; *a+bx+rnorm(length(x),s2)*) for each of the posterior draws. Estimate uncertainties of this ensemble. Compare these uncertainties to *predict(my.lm,interval='prediction')*.