**Testing uniform vs. normal Prior distribution**

1. First how we choose the next candidate parameter
2. Second how we calculate the prior probability density for the selected candidate
3. Choosing the candidate:

* We use multivariate normal jump away from the current point
* R function “rmvnorm” that generates random deviates using multivariate normal distribution with mean, µ and covariance matrix, . We define
* Now we choose a candidate based on

Metropolis Ratio (MR) =

* We directly accept the candidate if MR > 1.0
* Otherwise the candidate can still be accepted but only with the probability equal to MR

1. Calculating the prior probability density for the selected candidate:

* We use R function “dunif” that gives the density for uniform distribution on the interval from min to max
* The idea behind this is we don’t have any prior knowledge about the modelled parameters except the parameter space that we already defined
* So the final modelled parameters can fall anywhere within the parameter space and the candidate should have a uniform prior probability density within the assigned parameter space
* We then compare this result with normal prior distribution using another R function “dnorm” that gives the density for normal distribution with mean equal to mean and standard deviation equal to sd
* The comparison with all modelled parameters are given below in the table
* In this particular experiment, the final modelled parameters are quite similar whether we use uniform or normal prior probability density, and the possible explanation is the likelihood function has more weight compared to prior, and that minimizes the variation of prior probability density calculations
* But reducing standard deviation with normal distribution gives different outcomes from uniform priors

Consider two groups of treatments for testing the prior distribution (uniform vs. normal)

|  |  |
| --- | --- |
| **Prior uniform distribution** | **Prior normal (Gaussian) distribution** |
| Function: [dunif (x, min, max)] | Function: [dnorm (x, mean, sd)] |
| *We use: dunif (candidatepValues, pMinima, pMaxima)* | *We use: dnorm (candidatepValues, (pMinima + pMaxima)/2, 1.0)* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |