Prior Analysis Jose’s Data and Posteriors

Matt Wheeler

9/22/2020

## Sensitivity Analysis of priors

I am going to look at the different priors and the resulting likelihood that results for the following data-set using maximum likelihood estimation. Note that I have made all of the doses on a (0,1) scale for more appropriate comparisons.

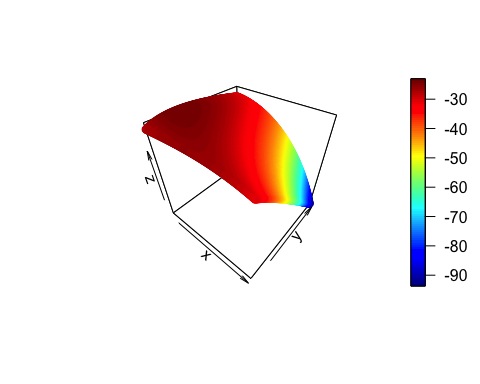
print(mData)

## [,1] [,2] [,3]  
## [1,] 0.000 1 10  
## [2,] 0.075 4 10  
## [3,] 0.250 4 10  
## [4,] 1.000 7 10

H = single\_dichotomous\_fit(mData[,1],mData[,2],mData[,3],model\_type ="weibull",fit\_type = "mle")  
H$parameters

## [1] -2.1666034 0.4443281 1.0068814

The default MLE for the parameter hits its natural bound at 1.0. If we look at the likelihood without the beta we have the following surface:

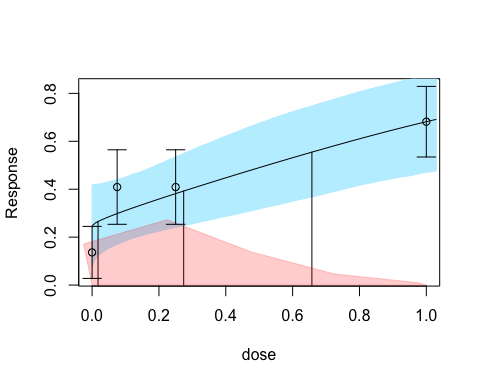


So what happens when we start to mess around with the prior? Well lets see the default prior in `ToxicR’ using MCMC for pretty plots we have:

H = single\_dichotomous\_fit(mData[,1],mData[,2],mData[,3],model\_type = "weibull",fit\_type = "mcmc")  
print( H$fitted\_model$parameters)

## [1] -1.0913760 1.0051454 0.7630374

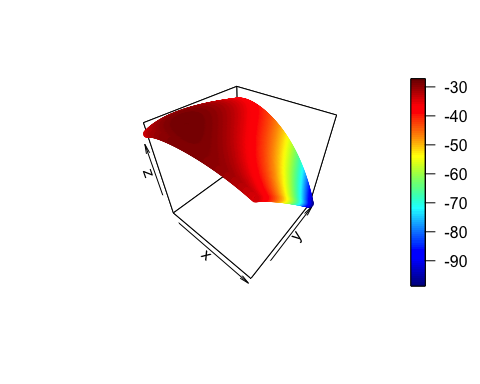
plot(H)



H$prior

## $prior  
## Model Parameter Priors  
## ------------------------------------------------------------------------  
## Prior: Normal(mu = 0.00, sd = 2.000) 1[-20.00,20.00]  
## Prior: Log-Normal(log-mu = 0.42, log-sd = 0.500) 1[0.00,40.00]  
## Prior: Log-Normal(log-mu = 0.00, log-sd = 1.500) 1[0.00,10000.00]

Now the parameters are essentially no different than the estimate using maximum likelihood, but the experimental test prior, which is different than BMDS is pretty flat. What does the optimization surface look like?

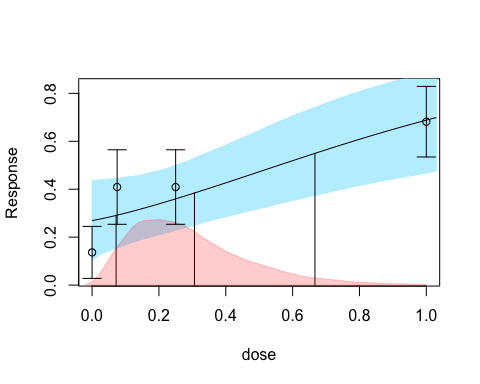
 The peak is not noticeably different, so it is essentially performing the same inference.

### Jose’s First Prior

prior <- create\_prior\_list(normprior(0,2,-20,20),  
 lnormprior(0.5,0.3,0,20),  
 lnormprior(0,1.5,0,10000))  
H = single\_dichotomous\_fit(mData[,1],mData[,2],mData[,3],model\_type = "weibull",fit\_type = "mcmc",prior = prior)  
print( H$fitted\_model$parameters)

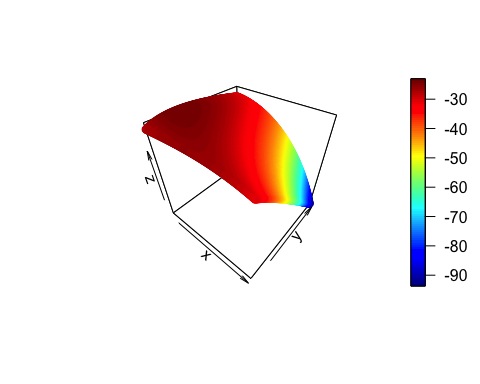
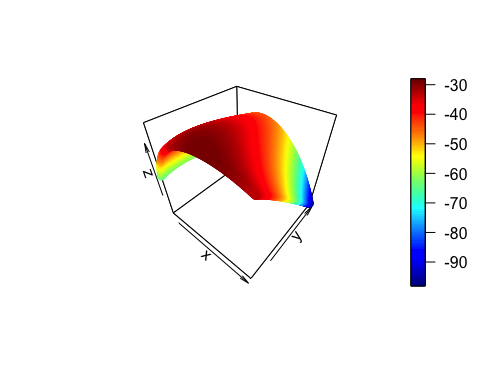
## [1] -0.9061123 1.4451849 0.6808757

plot(H)



H$prior

## $prior  
## Model Parameter Priors  
## ------------------------------------------------------------------------  
## Prior: Normal(mu = 0.00, sd = 2.000) 1[-20.00,20.00]  
## Prior: Log-Normal(log-mu = 0.50, log-sd = 0.300) 1[0.00,20.00]  
## Prior: Log-Normal(log-mu = 0.00, log-sd = 1.500) 1[0.00,10000.00]

Here the alpha kicks up to thus our prior is starting to impact our fit. What does this prior surface look like?  Here the top plot is the ‘new prior’ and the bottom plot is the likelihood, and this side-by-side view starts to show us what is going on. The bottom plot has no more than a 0.5 change in the log likelihood between the MLE and the edge of the graph on the x axis which is the shape parameter. The change is really insignificant, and the prior is thus informative increasing

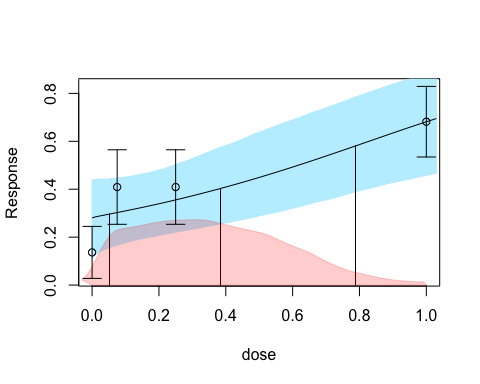
### Jose’s Second prior

Looking at the second prior on the list we can do the same as above.

prior <- create\_prior\_list(normprior(0,2,-20,20),  
 lnormprior(0.8,0.5,0,20),  
 lnormprior(0,1.5,0,10000))  
H = single\_dichotomous\_fit(mData[,1],mData[,2],mData[,3],model\_type = "weibull",fit\_type = "mcmc",prior = prior)  
print( H$fitted\_model$parameters)

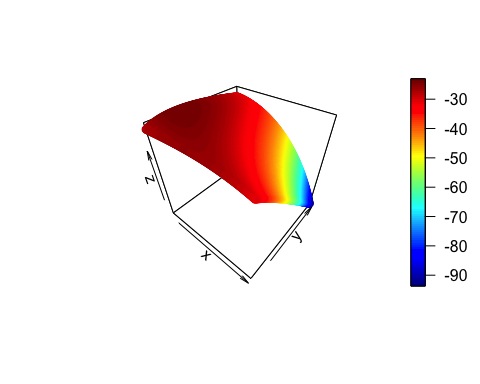
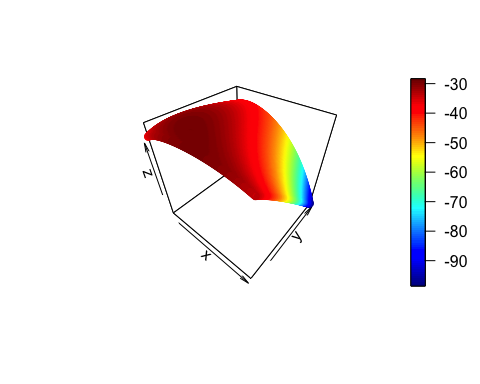
## [1] -0.8790296 1.5621925 0.6656322

plot(H)



H$prior

## $prior  
## Model Parameter Priors  
## ------------------------------------------------------------------------  
## Prior: Normal(mu = 0.00, sd = 2.000) 1[-20.00,20.00]  
## Prior: Log-Normal(log-mu = 0.80, log-sd = 0.500) 1[0.00,20.00]  
## Prior: Log-Normal(log-mu = 0.00, log-sd = 1.500) 1[0.00,10000.00]

 Given there is more variance the ellipsis/quadratic does not move as far but it still moves. The problem is the same as above. There is barely any difference between an and an as far as inference is concerned.

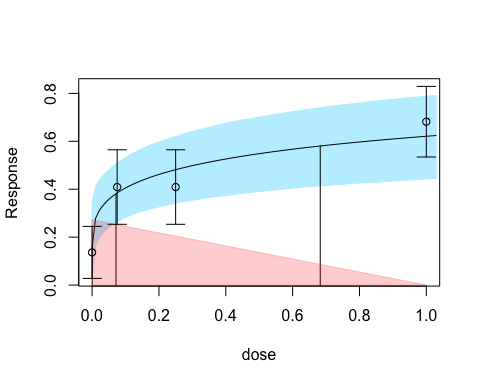
### Jose’s Third prior

Looking at the second prior on the list we can do the same as above.

prior <- create\_prior\_list(normprior(0,2,-20,20),  
 lnormprior(-1,0.2,0,20),  
 lnormprior(0,1.5,0,10000))  
H = single\_dichotomous\_fit(mData[,1],mData[,2],mData[,3],model\_type = "weibull",fit\_type = "mcmc",prior = prior)  
print( H$fitted\_model$parameters)

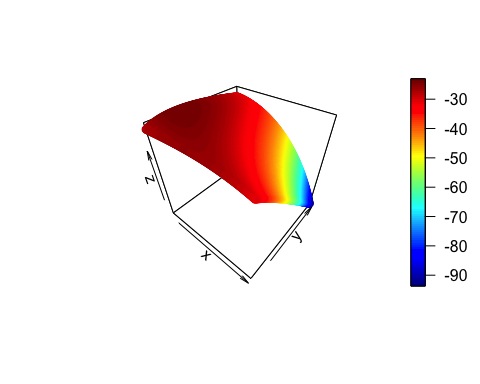
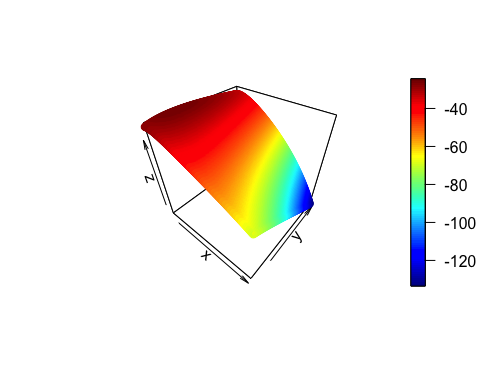
## [1] -1.6671302 0.3549661 0.7362700

plot(H)



H$prior

## $prior  
## Model Parameter Priors  
## ------------------------------------------------------------------------  
## Prior: Normal(mu = 0.00, sd = 2.000) 1[-20.00,20.00]  
## Prior: Log-Normal(log-mu = -1.00, log-sd = 0.200) 1[0.00,20.00]  
## Prior: Log-Normal(log-mu = 0.00, log-sd = 1.500) 1[0.00,10000.00]



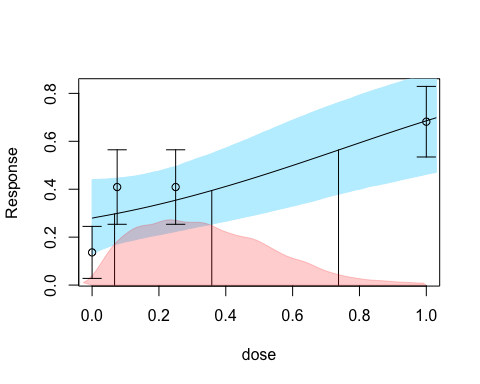
Now this prior is wacky, but it gives the result we would expect. It puts all of the mass of below 1 and give a nutjob (scientific term) surface to optimize.

### BMDS Standard Prior

prior <- create\_prior\_list(normprior(0,2,-20,20),  
 lnormprior(0.69,0.42,0,20),  
 lnormprior(0,1.5,0,10000))  
H = single\_dichotomous\_fit(mData[,1],mData[,2],mData[,3],model\_type = "weibull",fit\_type = "mcmc",prior = prior)  
print( H$fitted\_model$parameters)

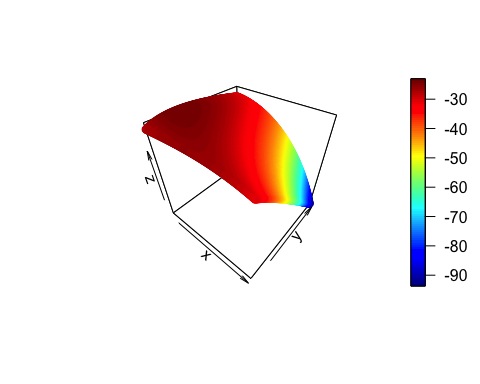
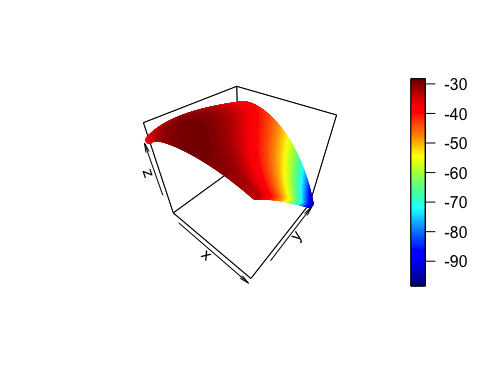
## [1] -0.8810978 1.5524636 0.6666730

plot(H)



H$prior

## $prior  
## Model Parameter Priors  
## ------------------------------------------------------------------------  
## Prior: Normal(mu = 0.00, sd = 2.000) 1[-20.00,20.00]  
## Prior: Log-Normal(log-mu = 0.69, log-sd = 0.420) 1[0.00,20.00]  
## Prior: Log-Normal(log-mu = 0.00, log-sd = 1.500) 1[0.00,10000.00]



Again, we see that the default BMDS prior puts things in a much more ‘reasonable’ area, because truth be told there is not a heck of a lot of information to put below zero. In fact a likelihood difference of 0.5 is barely a sneeze. To me, this just shows how little information can be in these data sets that inform our models. As always, being careful is the name of the game.

### Continuous Model Averaging

Jay here is Continuous MA

library(ToxicR)  
  
set.seed(893223)  
  
D <-c(rep(seq(0,1.0,1/4),each=4))  
mean <- 2.3 + 10/(1+exp(-(D-0.60)\*8))\*(1/(1+exp(-(0.99-D)\*13)))  
  
Y <- mean + rnorm(length(mean),0,0.7)  
#Q <- single\_continuous\_fit(as.matrix(D),as.matrix(Y),sstat = F,BMR = 1.0 ,model\_type="FUNL",distribution = "normal",fit\_type = "laplace")  
  
system.time({fit<-ma\_continuous\_fit(D,Y,fit\_type="mcmc",samples=25000,burnin=2500,BMR=2.0)})

## user system elapsed   
## 13.758 0.263 1.194

system.time({fit2<-ma\_continuous\_fit(D,Y,fit\_type="laplace",BMR=2.0)})

## user system elapsed   
## 8.852 0.487 1.014

plot(fit)

