

site in phi

Timothée Bonnet

24 June 2015

1. Attention au logit / logit^{-1}

Je ne suis pas sûr de comprendre pourquoi tu calcules ça:

```
phi.site[u] <- 1 / (1+exp (-mean.gamma[u]))
```

Ça ne te donne **pas** la survie moyenne du site u, ni la difference entre u et les autres sites, parce que l'intercept ou le sexe ne sont pas inclus dans ta transformation, et que la transformation n'est pas linéaire.

Par exemple, si mean.gamma=0.5, en fonction de l'intercept, tu peux avoir différentes probabilités de survie, et ces probabilités ne sont pas espacées linéairement:

```
par(mfrow=c(1,2))
curve(expr = 1/(1+exp(-x)),from = -10,to=10, las=1,ylab="p(survie)",xlab="logit scale",main="P(survie) )

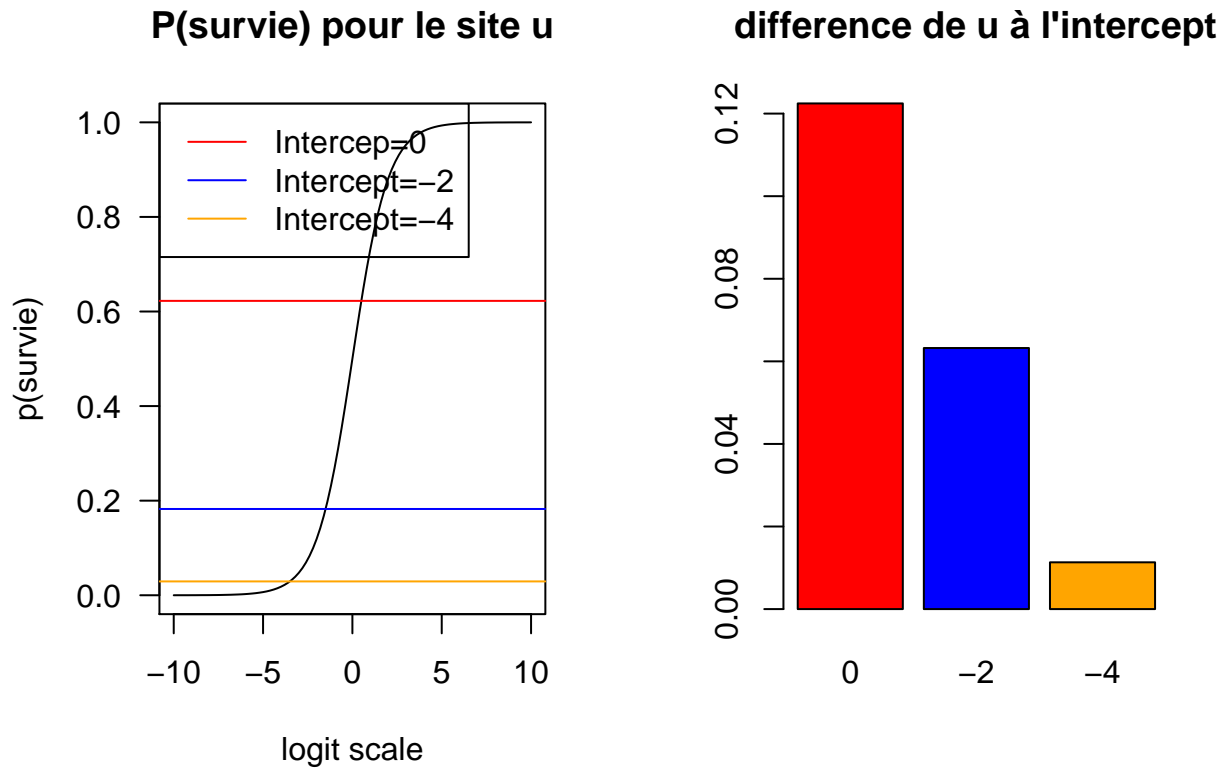
mean.gamma<-0.5
abline(h = 1/(1+exp(-mean.gamma)),col="red")

mean.gamma<-0.5
alpha<- -2
abline(h = 1/(1+exp(-(alpha+mean.gamma))),col="blue")

mean.gamma<-0.5
alpha<- -4
abline(h = 1/(1+exp(-(alpha+mean.gamma))),col="orange")

legend(x = "topleft",legend = c("Intercep=0", "Intercept=-2","Intercept=-4"),col=c("red","blue","orange"))

barplot(main="difference de u à l'intercept",height = c(1/(1+exp(-(0+0.5)))-1/(1+exp(-(0))), 1/(1+exp(-(0+0.5)))-1/(1+exp(-(0))), 1/(1+exp(-(0+0.5)))-1/(1+exp(-(0)))
```



2. Ma proposition de modèle pour introduire site dans la survie

En supposant qu'un même individu ne peut être observé que sur un site...

```
#Model
for(i in 1:I) #boucle sur les individus
{
  for(t in f[i):(Kmax-1)) #boucle sur le temps
  {
    logit(phi[i,t]) <- alpha[t] + phi.sex[sex[i]]
    logit(p[i,t]) <- beta[t] + p.sex[sex[i]] + Randomsite[site[i]]
  } #t
} #i

#Priors

#for survival parameters
for(t in 1:(Kmax-1)){
  alpha[t] ~ dnorm(0,0.01)I(-10,10)
  #phi.sex1[t]<-1 / (1 + exp(-alpha [t]))
  #phi.sex2[t]<-1 / (1 + exp(-alpha [t] - phi.sex[2])) Inutile, ça n'aide pas à l'estimation et coûte
}

phi.sex[1]<-0
```

```

phi.sex[2] ~ dnorm(0,0.01)I(-10,10)

#for recapture parameters
for(t in 1:(Kmax-1)){
  beta[t] ~ dnorm(0,0.01)I(-10,10)
  #p.sex1[t]<-1 / (1 + exp(-beta [t]))
  #p.sex2[t]<-1 / (1 + exp(-beta [t] - p.sex[2])) Inutile, ça n'aide pas à l'estimation et coute du t
}

p.sex[1]<-0
p.sex[2] ~ dnorm(0,0.01)I(-10,10)

for(u in 1:nsite){

  Randomsite[u] ~ dnorm(0, tausite)#random effect site

  #gamma[u] ~ dnorm(mean.gamma, tau)
  #phi.site[u] <- 1 / (1+exp (-mean.gamma[u]))
}
#mean.gamma ~ dnorm (0, 0.001)
#sigma      ~ dunif(0,10)
# tau<-pow(sigma, -2)

tausite<-pow(sigmasite,-2)#random precision site
sigmasite~dunif(0,10)#random sd site

```