**##power of the study under H0 model : ( p(c\*t), theta(t), lambda(.) ) and**

**Ha model : ( p(c\*t), theta(t), lambda(c) )**

# give the categories in the population

categories = c("M","F")

# give the different capture probabilities for p1M, p1F, p2M and p2F

cap.prob = c( 0.05, 0.08, 0.1, 0.12)

# give the category proportions( total should add up to 1)

lambda= c(0.4, 0.6)

# give the subsample proportions for the time 1 and 2

theta = c(0.8, 0.5)

# total numbers individuals capture for the study

sample.size = 2000

n.simulation = 1000 # number of simulations to verify the power of the study.

###### alternative (unrestricted) model #####################################

#model identification : unrestricted model

unrest.model.id = paste("{ p(c\*t), theta(t), lambda(c) }")

# give the required design matrices

unrest.captureDM = create.DM(c(1,2,3,4)) # Design matrix for capture recapture probabilities

unrest.thetaDM = create.DM(c(1,2)) # Design matrix for theta(sampling(sexing) fractions)

unrest.lambdaDM = create.DM(c(1)) # Design matrix for lambda(Category proportion)

#give the offset vectors(vectors of zeros should be given since no restriction)

unrest.captureOFFSET = c(0,0,0,0)

unrest.thetaOFFSET = c(0,0)

unrest.lambdaOFFSET = c(0)

##### null (restricted) model ##########################################

#model identification : restricted model(capture probabilities equal)

rest.model.id = paste("{ p(t), theta(t), lambda(0.4+Delta) }")

# Design matrices restricted model

rest.captureDM = create.DM(c(1,2,3,4)) # capture probabilities are equal

rest.thetaDM = create.DM(c(1,2))

rest.lambdaDM = matrix(, ncol=0,nrow=1)

#offset vectors for restricted model

rest.captureOFFSET = c(0,0,0,0)

rest.thetaOFFSET = c(0,0)

rest.lambdaOFFSET = c(logit(0.4))Calculate the power for different increment of lambda values

Consider lambda\_M for null model: 0. 4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7

Then the Delta : 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3

|  |  |  |
| --- | --- | --- |
| Delta\_lambda | power\_n\_2000\_Devineau | power\_n\_2000\_simulation |
| 0 | 0.05 | 0.048 |
| 0.05 | 0.125 | 0.126 |
| 0.1 | 0.338 | 0.332 |
| 0.15 | 0.675 | 0.612 |
| 0.2 | 0.837 | 0.816 |
| 0.25 | 0.952 | 0.928 |
| 0.3 | 0.99 | 0.98 |
| 0.35 | 1 | 0.99 |
| 0.4 | 1 | 1 |

