**Generate data with capture heterogeneity and analysis of data**

###############################################################################

###### Identify the violation of assumption through the gof plots ############

######## Generate data with violation of assumptions ( Heterogeneity) #########

###############################################################################

# generate some data where the assumptions of the model are violated and see

# if the residual plot and the p-value of the gof test assessment capture the

# problem ( violation of assumptions : failure of non-death assumption, entering

# animals to the sampled population between the first and second period, etc..)

#

# phi = probability that animal alive at the time of the first sampling

# occasion is still alive and present in the population at the time of

# the second sampling occasion

# lambda\_B = fraction of the total net births that enter the system between

# time 1 and time 2

#

# p1 - capture probability at time 1

# p2 - capture probability at time 2

########################################################

source("load.R") # load required functions and packages

options(width=350)

set.seed(123)

########################################################

#### consider 2 male groups and two female groups ######

########################################################

N <- 100000 # population size

theta <- c(0.8,0.6) # sub sample proportions

phi <- .9

lambda\_B <- 1.e-10 # since this is converted to log form, have to give

# a very small number instead of zero

##################################

########## Male group 1 ##########

##################################

cat1 <- "M" # category

M\_lambda\_1 <- 0.3 # category proportion for a particular group of males

## give the mean, standard deviation and correlation between each pair of ###

## variables (p1, p2, phi, lambda\_B) for generating data with heterogeneity ###

### mean for p1, p2, phi, lambda\_B ###

p1 <- 0.2

p2 <- 0.3

mu <- c(p1, p2, phi, lambda\_B) # mean values as a vector

### standard deviation of each variable ###

s\_p1 <- 0.03 # standard deviation of p1 at time 1

s\_p2 <- 0.06 # standard deviation of p2 at time 2

s\_phi <- 0 # standard deviation of phi

s\_lambda\_B <- 0 # standard deviation of lambda\_B

### correlation between each pair of variables ####

R\_p1\_p2 <- 0.7 # correlation between p1 and p2

R\_p1\_phi <- 0 # correlation between p1 and phi

R\_p1\_lambda\_B <- 0 # correlation between p1 and lambda\_B

R\_p2\_phi <- 0 # correlation between p2 and phi

R\_p2\_lambda\_B <- 0 # correlation between p2 and lambda\_B

R\_phi\_lambda\_B <- 0 # correlation between phi and lambda\_B

## correlation matrix ##

R <- matrix(c( 1, R\_p1\_p2, R\_p1\_phi, R\_p1\_lambda\_B,

R\_p1\_p2, 1, R\_p2\_phi, R\_p2\_lambda\_B,

R\_p1\_phi, R\_p2\_phi, 1, R\_phi\_lambda\_B,

R\_p1\_lambda\_B,R\_p2\_lambda\_B,R\_phi\_lambda\_B, 1), ncol = 4)

## vector with standard deviation of each variable ##

standard\_dev <- c(s\_p1, s\_p2, s\_phi, s\_lambda\_B)

D <- diag(standard\_dev) # create a diagonal matrix

## convert correlation matrix to variance-covariance matrix ##

VCV <- D%\*%R%\*%D

input1 <-list(N=N,mu=mu,VCV=VCV,cat=cat1,theta=theta,lambda=M\_lambda\_1)

##################################

########## Male group 2 ##########

##################################

cat1 <- "M" # category

M\_lambda\_2 <- 0.2 # category proportion for a particular group of males

## give the mean, standard deviation and correlation between each pair of ###

## variables (p1, p2, phi, lambda\_B) for generating data with heterogeneity ###

### mean for p1, p2, phi, lambda\_B ###

p1 <- 0.3

p2 <- 0.4

mu <- c(p1, p2, phi, lambda\_B) # mean values as a vector

### standard deviation of each variable ###

s\_p1 <- 0.05 # standard deviation of p1 at time 1

s\_p2 <- 0.03 # standard deviation of p2 at time 2

s\_phi <- 0 # standard deviation of phi

s\_lambda\_B <- 0 # standard deviation of lambda\_B

### correlation between each pair of variables ####

R\_p1\_p2 <- 0.8 # correlation between p1 and p2

R\_p1\_phi <- 0 # correlation between p1 and phi

R\_p1\_lambda\_B <- 0 # correlation between p1 and lambda\_B

R\_p2\_phi <- 0 # correlation between p2 and phi

R\_p2\_lambda\_B <- 0 # correlation between p2 and lambda\_B

R\_phi\_lambda\_B <- 0 # correlation between phi and lambda\_B

## correlation matrix ##

R <- matrix(c( 1, R\_p1\_p2, R\_p1\_phi, R\_p1\_lambda\_B,

R\_p1\_p2, 1, R\_p2\_phi, R\_p2\_lambda\_B,

R\_p1\_phi, R\_p2\_phi, 1, R\_phi\_lambda\_B,

R\_p1\_lambda\_B,R\_p2\_lambda\_B,R\_phi\_lambda\_B, 1), ncol = 4)

## vector with standard deviation of each variable ##

standard\_dev <- c(s\_p1, s\_p2, s\_phi, s\_lambda\_B)

D <- diag(standard\_dev) # create a diagonal matrix

## convert correlation matrix to variance-covariance matrix ##

VCV <- D%\*%R%\*%D

input2 <- list(N=N,mu=mu,VCV=VCV,cat=cat1,theta=theta,lambda=M\_lambda\_2)

#########################

#### Female group 1 ####

#########################

cat2 <- "F" # category

F\_lambda\_1 <- 0.4 # category proportion for a particular group of females

## give the mean, standard deviation and correlation between each pair of ###

## variables (p1, p2, phi, lambda\_B) for generating data with heterogeneity ###

### mean for p1, p2, phi, lambda\_B ###

p1 <- 0.2

p2 <- 0.4

mu <- c(p1, p2, phi, lambda\_B) # mean values as a vector

### standard deviation of each variable ###

s\_p1 <- 0.01 # standard deviation of p1 at time 1

s\_p2 <- 0.05 # standard deviation of p2 at time 2

s\_phi <- 0 # standard deviation of phi

s\_lambda\_B <- 0 # standard deviation of lambda\_B

### correlation between each pair of variables ####

R\_p1\_p2 <- 0.6 # correlation between p1 and p2

R\_p1\_phi <- 0 # correlation between p1 and phi

R\_p1\_lambda\_B <- 0 # correlation between p1 and lambda\_B

R\_p2\_phi <- 0 # correlation between p2 and phi

R\_p2\_lambda\_B <- 0 # correlation between p2 and lambda\_B

R\_phi\_lambda\_B <- 0 # correlation between phi and lambda\_B

## correlation matrix ##

R <- matrix(c( 1, R\_p1\_p2, R\_p1\_phi, R\_p1\_lambda\_B,

R\_p1\_p2, 1, R\_p2\_phi, R\_p2\_lambda\_B,

R\_p1\_phi, R\_p2\_phi, 1, R\_phi\_lambda\_B,

R\_p1\_lambda\_B,R\_p2\_lambda\_B,R\_phi\_lambda\_B, 1), ncol = 4)

## vector with standard deviation of each variable ##

standard\_dev <- c(s\_p1, s\_p2, s\_phi, s\_lambda\_B)

D <- diag(standard\_dev) # create a diagonal matrix

## convert correlation matrix to variance-covariance matrix ##

VCV <- D%\*%R%\*%D

input3 <- list(N=N,mu=mu,VCV=VCV,cat=cat2,theta=theta,lambda=F\_lambda\_1)

#########################

#### Female group 2 ####

#########################

cat2 <- "F" # category

F\_lambda\_2 <- 0.1 # category proportion for a particular group of females

## give the mean, standard deviation and correlation between each pair of ###

## variables (p1, p2, phi, lambda\_B) for generating data with heterogeneity ###

### mean for p1, p2, phi, lambda\_B ###

p1 <- 0.3

p2 <- 0.3

mu <- c(p1, p2, phi, lambda\_B) # mean values as a vector

### standard deviation of each variable ###

s\_p1 <- 0.02 # standard deviation of p1 at time 1

s\_p2 <- 0.06 # standard deviation of p2 at time 2

s\_phi <- 0 # standard deviation of phi

s\_lambda\_B <- 0 # standard deviation of lambda\_B

### correlation between each pair of variables ####

R\_p1\_p2 <- 0.5 # correlation between p1 and p2

R\_p1\_phi <- 0 # correlation between p1 and phi

R\_p1\_lambda\_B <- 0 # correlation between p1 and lambda\_B

R\_p2\_phi <- 0 # correlation between p2 and phi

R\_p2\_lambda\_B <- 0 # correlation between p2 and lambda\_B

R\_phi\_lambda\_B <- 0 # correlation between phi and lambda\_B

## correlation matrix ##

R <- matrix(c( 1, R\_p1\_p2, R\_p1\_phi, R\_p1\_lambda\_B,

R\_p1\_p2, 1, R\_p2\_phi, R\_p2\_lambda\_B,

R\_p1\_phi, R\_p2\_phi, 1, R\_phi\_lambda\_B,

R\_p1\_lambda\_B,R\_p2\_lambda\_B,R\_phi\_lambda\_B, 1), ncol = 4)

## vector with standard deviation of each variable ##

standard\_dev <- c(s\_p1, s\_p2, s\_phi, s\_lambda\_B)

D <- diag(standard\_dev) # create a diagonal matrix

## convert correlation matrix to variance-covariance matrix ##

VCV <- D%\*%R%\*%D

input4 <- list(N=N,mu=mu,VCV=VCV,cat=cat2,theta=theta,lambda=F\_lambda\_2)

###################################################################

#### check whether the proportion for each category is correct ####

###################################################################

tot\_prop <- M\_lambda\_1 + M\_lambda\_2 + F\_lambda\_1 + F\_lambda\_2

if((tot\_prop != 1)){stop("Error: proportion for each category is wrong.

Total does not equal to 1")}

###############################################################################

##### fit the model for the generated data(with heterogeneity) with the #######

##### following design matrices and offset vectors ############################

#model identification : unrestricted model

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

captureDM <- create.DM(c(1,2,3,4))

thetaDM <- create.DM(c(1,2))

lambdaDM <- create.DM(c(1))

#give the offset vectors(vectors of zero's should be given since no restriction)

captureOFFSET <- c(0,0,0,0)

thetaOFFSET <- c(0,0)

lambdaOFFSET <- c(0)

###############################################################################

############### fitting the model to data with heterogeneity ##################

# create list of list as input data

# there are 4 list as input1,input2,input3,input4 since 2 male groups and two female groups

input =list(input1,input2,input3,input4)

#################################################################

## analysis and residual plot for one data set with heterogeneity

gen.data.set <- ldply(input, generate.data.heterogeneity)

data <- NULL

data$History <- as.vector(gen.data.set$History)

data$counts <- gen.data.set$counts

data$category <- c(cat1,cat2)

model\_res <- fit.model(model.id,data,captureDM,thetaDM,lambdaDM,

captureOFFSET,thetaOFFSET,lambdaOFFSET)

# print results (Model information, MLEs, SE,residual plot, etc..)

print.output(model\_res)

##################################################################

# "n.sim" is the number of simulations of heterogeneity data sets

# simulate "n.sim" number of data set with heterogeneity and fit the model

# to each data set and save the estimates in a data frame and then calculate

# the mean vales of the estimates

#histogram of estimates of population size and the mean values of the all estimates

# the function "simulation.results" is in the file "generate.data.heterogeneity.R"

heterogeneity.results <- simulation.results(n.sim=1000,input,model.id,

captureDM,thetaDM,lambdaDM,

captureOFFSET,thetaOFFSET,lambdaOFFSET)

heterogeneity.results

###############################################################################

###############################################################################

########## check the code using simple Lincoln Petersen estimates #############

###############################################################################

N <- 100000 # population size

theta <- c(1,1) # sub sample proportions

phi <- .99999999 # Since this is converted to logit, give a value

# closer to 1 instead of value 1

lambda\_B <- 1.e-10 # since this is converted to log form, have to give

# a very small number instead of zero

cat1 <- "M" # category

lambda\_M <- 0.4 # category proportion for a particular group of males

## give the mean, standard deviation and correlation between each pair of ###

## variables (p1, p2, phi, lambda\_B) for generating data with heterogeneity ###

### mean for p1, p2, phi, lambda\_B ###

p1 <- 0.2

p2 <- 0.3

mu <- c(p1, p2, phi, lambda\_B) # mean values as a vector

### standard deviation of each variable ###

s\_p1 <- 0 # standard deviation of p1 at time 1

s\_p2 <- 0 # standard deviation of p2 at time 2

s\_phi <- 0 # standard deviation of phi

s\_lambda\_B <- 0 # standard deviation of lambda\_B

### correlation between each pair of variables ####

R\_p1\_p2 <- 0 # correlation between p1 and p2

R\_p1\_phi <- 0 # correlation between p1 and phi

R\_p1\_lambda\_B <- 0 # correlation between p1 and lambda\_B

R\_p2\_phi <- 0 # correlation between p2 and phi

R\_p2\_lambda\_B <- 0 # correlation between p2 and lambda\_B

R\_phi\_lambda\_B <- 0 # correlation between phi and lambda\_B

## correlation matrix ##

R <- matrix(c( 1, R\_p1\_p2, R\_p1\_phi, R\_p1\_lambda\_B,

R\_p1\_p2, 1, R\_p2\_phi, R\_p2\_lambda\_B,

R\_p1\_phi, R\_p2\_phi, 1, R\_phi\_lambda\_B,

R\_p1\_lambda\_B,R\_p2\_lambda\_B,R\_phi\_lambda\_B, 1), ncol = 4)

## vector with standard deviation of each variable ##

standard\_dev <- c(s\_p1, s\_p2, s\_phi, s\_lambda\_B)

D <- diag(standard\_dev) # create a diagonal matrix

## convert correlation matrix to variance-covariance matrix ##

VCV <- D%\*%R%\*%D

input\_M <-list(N=N,mu=mu,VCV=VCV,cat=cat1,theta=theta,lambda=lambda\_M)

###########################################

cat2 <- "F" # category

lambda\_F <- 0.6 # category proportion for a particular group of males

## give the mean, standard deviation and correlation between each pair of ###

## variables (p1, p2, phi, lambda\_B) for generating data with heterogeneity ###

### mean for p1, p2, phi, lambda\_B ###

p1 <- 0.2

p2 <- 0.3

mu <- c(p1, p2, phi, lambda\_B) # mean values as a vector

### standard deviation of each variable ###

s\_p1 <- 0 # standard deviation of p1 at time 1

s\_p2 <- 0 # standard deviation of p2 at time 2

s\_phi <- 0 # standard deviation of phi

s\_lambda\_B <- 0 # standard deviation of lambda\_B

### correlation between each pair of variables ####

R\_p1\_p2 <- 0 # correlation between p1 and p2

R\_p1\_phi <- 0 # correlation between p1 and phi

R\_p1\_lambda\_B <- 0 # correlation between p1 and lambda\_B

R\_p2\_phi <- 0 # correlation between p2 and phi

R\_p2\_lambda\_B <- 0 # correlation between p2 and lambda\_B

R\_phi\_lambda\_B <- 0 # correlation between phi and lambda\_B

## correlation matrix ##

R <- matrix(c( 1, R\_p1\_p2, R\_p1\_phi, R\_p1\_lambda\_B,

R\_p1\_p2, 1, R\_p2\_phi, R\_p2\_lambda\_B,

R\_p1\_phi, R\_p2\_phi, 1, R\_phi\_lambda\_B,

R\_p1\_lambda\_B,R\_p2\_lambda\_B,R\_phi\_lambda\_B, 1), ncol = 4)

## vector with standard deviation of each variable ##

standard\_dev <- c(s\_p1, s\_p2, s\_phi, s\_lambda\_B)

D <- diag(standard\_dev) # create a diagonal matrix

## convert correlation matrix to variance-covariance matrix ##

VCV <- D%\*%R%\*%D

input\_F <-list(N=N,mu=mu,VCV=VCV,cat=cat2,theta=theta,lambda=lambda\_F)

#### fit the model for the generated data(no heterogeneity in the data) #################

#model identification : unrestricted model

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

captureDM <- create.DM(c(1,1,2,2))

thetaDM <- create.DM(c(1,2))

lambdaDM <- create.DM(c(1))

#give the offset vectors(vectors of zero's should be given since no restriction)

captureOFFSET <- c(0,0,0,0)

thetaOFFSET <- c(0,0)

lambdaOFFSET <- c(0)

#########################################

# create list of list as input data

input =list(input\_M,input\_F)

n.sim =1000 #the number of simulations of heterogeneity data sets

LP.sim.data <- rdply(n.sim,fit.model.to.heterogeneity.data(input,model.id,

captureDM,thetaDM,lambdaDM,

captureOFFSET,thetaOFFSET,lambdaOFFSET))

names(LP.sim.data) <- c("sample","N","N\_M","N\_F", "p\_1M","p\_1F", "p\_2M",

"p\_2F", "lambda\_M", "lambda\_F", "theta\_1", "theta\_2")

# mean of the estimates of the population size is displyed inside the graph

x.pos = min(LP.sim.data$N) +

0.75\*(max(LP.sim.data$N) - min(LP.sim.data$N))

#histogram for the estimates of the population size

histogram\_N\_hat\_LP <- ggplot(LP.sim.data, aes(x=N,..density.. ))+

geom\_histogram(aes(y=..density.. ))+

xlab("Estimates of N") + ylab("Density")+

ggtitle(paste("Histogram of population estimates for non-heterogeneity Data",

"\n model: ", model.id,

"\n number of simulation = ", n.sim, ", initial pop size = N =",N, sep=" "))+

theme(axis.text=element\_text(size=10,face="bold"),

axis.title=element\_text(size=14,face="bold"),

plot.title = element\_text(size = 14,face="bold"))+

annotate("text", label = paste("mean = ",

round(mean(LP.sim.data$N)),sep=""),

x = x.pos, hjust = 0, y = Inf, vjust = 2, color = "darkred")

N\_M <- N \* lambda\_M

N\_F <- N \* lambda\_F

p\_1M <- p1

p\_1F <- p1

p\_2M <- p2

p\_2F <- p2

initial.value <- prettyNum(round(c(N,N\_M,N\_F,p\_1M,p\_1F,p\_2M,p\_2F, lambda\_M,

lambda\_F, theta),3),, big.mark = ",")

names(initial.value) <- c("N","N\_M","N\_F", "p\_1M","p\_1F", "p\_2M",

"p\_2F", "lambda\_M", "lambda\_F", "theta\_1", "theta\_2")

sim.means.LP <- prettyNum(round(colMeans(LP.sim.data[2:ncol(LP.sim.data)]),3),

big.mark = ",")

df.initial\_sim.means <- as.data.frame(cbind(initial.value,sim.means.LP))

names(df.initial\_sim.means) <- c( "initial.values" ,"mean\_estimate")

#histogram for the estimates of the population size

histogram\_N\_hat\_LP

# initial parameter values and mean of the estimates using simulations

df.initial\_sim.means

###############################################################################

######## end of check the code using simple Lincoln Petersen estimates ########

###############################################################################

**Analysis and residual plot for one data set with capture heterogeneity**

Initial values used for optimization routine:

Initial capture probabilities:

time1 time2

M 0.2350288 0.3320703

F 0.2350288 0.3320703

Initial category proportions:

lambda\_ M lambda\_ F

0.5 0.5

Initial sub-sample proportions:

theta\_1 theta\_2

0.7989580 0.6083423

Initial population size for optimization(simple Lincoln Petersen estimator is used) : 98,813

Design matrix and OFFSET vector for capture probabilities:

Beta.1 Beta.2 Beta.3 Beta.4 OFFSET.vector

p1M 1 0 0 0 0

p1F 0 1 0 0 0

p2M 0 0 1 0 0

p2F 0 0 0 1 0

Design matrix and OFFSET vector for sub-sample proportions (theta):

Beta.1 Beta.2 OFFSET.vector

theta\_1 1 0 0

theta\_2 0 1 0

Design matrix and OFFSET vector for category proportions (lambda):

Beta OFFSET.vector

lambda\_M 1 0

Find MLEs:

MLEs for capture probabilities:

time1 time2

M 0.2608458 0.3297775

F 0.2119846 0.3346467

MLEs for category proportions:

lambda\_ M lambda\_ F

0.4754415 0.5245585

MLEs for sub-sample proportions:

theta\_1 theta\_2

0.7989581 0.6083423

MLE for Population size : 98,736

MLE for Population size of category M : 46,943

MLE for Population size of category F : 51,793

SE's of the MLEs

SE's of the MLEs of capture probabilities:

time1 time2

M 0.003829928 0.004491420

F 0.003348127 0.004786395

SE's of the MLEs of the category proportions:

lambda\_ M lambda\_ F

0.005106391 0.005106391

SE's of the MLEs of the sub-sample proportions:

theta\_1 theta\_2

0.002629885 0.003080930

SE of the MLE of the population size: 808

SE for Population size of category M : 597

SE for Population size of category F : 696

Observed and Expected counts for capture histories for the model { p(c\*t), theta(t), lambda(c) }

History Observed.Counts Expected.counts Residual Standardized.Residuals

1 U0 3139 3118.537 20.4631716 0.372363144

2 UU 1530 1550.487 -20.4865934 -0.524412155

3 M0 6546 6556.891 -10.8914632 -0.139206270

4 F0 5827 5836.447 -9.4473948 -0.127487844

5 MM 3237 3226.265 10.7349155 0.192159817

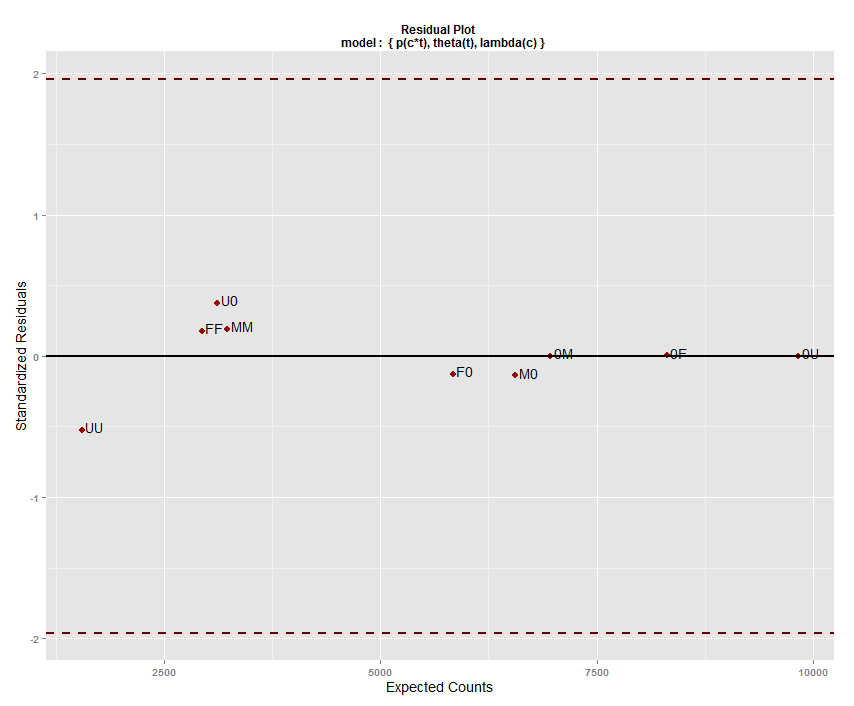
6 FF 2945 2935.505 9.4953816 0.177920073

7 0M 6961 6961.058 -0.0575603 -0.000715585

8 0F 8309 8308.780 0.2201332 0.002523511

9 0U 9831 9830.894 0.1063846 0.001130724

Standardized residual plot for the model { p(c\*t), theta(t), lambda(c) }



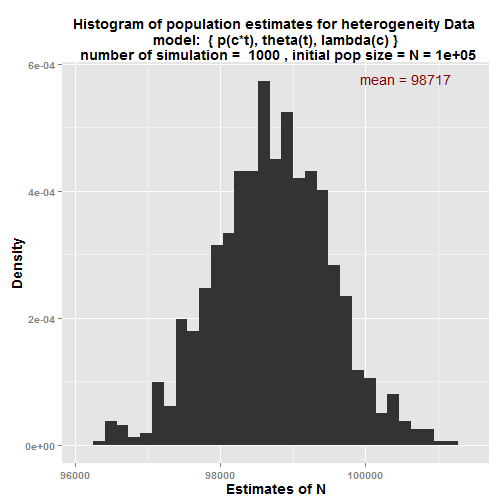
N <- 100000 # population size

theta <- c(0.8,0.6) # sub sample proportions

phi <- .9

lambda\_B <- 1.e-10 # since this is converted to log form, have to give

# a very small number instead of zero



$sim.means

mean\_estimate

N 98,717

N\_M 47,965

N\_F 50,752

p\_1M 0.252

p\_1F 0.217

p\_2M 0.321

p\_2F 0.338

lambda\_M 0.486

lambda\_F 0.514

theta\_1 0.8

theta\_2 0.6

**Check the code using simple Lincoln Petersen estimates**

N <- 100000 # population size

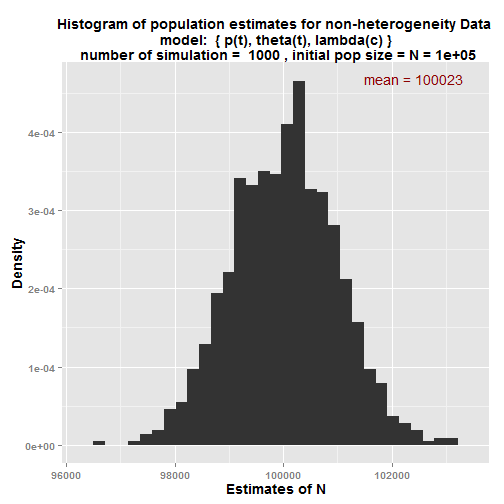
theta <- c(1,1) # sub sample proportions

phi <- .99999999 # Since this is converted to logit, give a value

# closer to 1 instead of value 1

lambda\_B <- 1.e-10 # since this is converted to log form, have to give

# a very small number instead of zero



# initial parameter values and mean of the estimates using simulations

df.initial\_sim.means

initial.param.values mean\_estimate

N 1e+05 1e+05

N\_M 40,000 40,011

N\_F 60,000 60,012

p\_1M 0.2 0.2

p\_1F 0.2 0.2

p\_2M 0.3 0.3

p\_2F 0.3 0.3

lambda\_M 0.4 0.4

lambda\_F 0.6 0.6

theta\_1 1 1

theta\_2 1 1