**Exercises 4 - 5**

In exercises 4 and 5 we are going to introduce the concept of conditional dependence between tests. For a better understanding of the training material please have a close look at the:

* related reading material titled "3.BLCM Assumptions\_LMessam"
* the following references:
* Bayesian Approaches to Modeling the Conditional Dependence Between Multiple Diagnostic Tests - Nandini Dendukuri (2004) <https://doi.org/10.1111/j.0006-341X.2001.00158.x>
* On the interpretation of test sensitivity in the two-test two-population problem: Assumptions matter - Wesley O.Johnson (2009) <https://doi.org/10.1016/j.prevetmed.2009.06.006>

**Exercise 4 - Three tests without conditional dependence between tests**

1. Load the dataset that is as a csv file (ex45\_data.csv) in your working directory

df <- read.csv("ex45\_data.csv")

#install.packages("dplyr")

library(dplyr)

1. Create a contingency table for each population

###contingency tables

df %>% mutate(x111 = ifelse(test1==1 & test2==1 & test3==1, 1, 0),

x100 = ifelse(test1==1 & test2==0 & test3==0, 1, 0),

x010 = ifelse(test1==0 & test2==1 & test3==0, 1, 0),

x001 = ifelse(test1==0 & test2==0 & test3==1, 1, 0),

x110 = ifelse(test1==1 & test2==1 & test3==0, 1, 0),

x101 = ifelse(test1==1 & test2==0 & test3==1, 1, 0),

x011 = ifelse(test1==0 & test2==1 & test3==1, 1, 0),

x000 = ifelse(test1==0 & test2==0 & test3 == 0, 1, 0)) %>%

group\_by(Population) %>%

summarise(x111=sum(x111), x100=sum(x100), x010=sum(x010),

x001=sum(x001), x110=sum(x110), x101=sum(x101),

x011=sum(x011), x000=sum(x000))

1. Write the model and save as “3\_test\_mod.bug” three tests – cell probabilities:

model{

##population 1

p1[1] <- pi1\*se1\*se2\*se3 + (1-pi1)\*(1-sp1)\*(1-sp2)\*(1-sp3) ###111

p1[2] <- pi1\*se1\*(1-se2)\*(1-se3) + (1-pi1)\*(1-sp1)\*sp2\*sp3 ###100

p1[3] <- pi1\*(1-se1)\*se2\*(1-se3) + (1-pi1)\*sp1\*(1-sp2)\*sp3 ###010

p1[4] <- pi1\*(1-se1)\*(1-se2)\*se3 + (1-pi1)\*sp1\*sp2\*(1-sp3) ##001

p1[5] <- pi1\*se1\*se2\*(1-se3) + (1-pi1)\*(1-sp1)\*(1-sp2)\*sp3 ###110

p1[6] <- pi1\*se1\*(1-se2)\*se3 + (1-pi1)\*(1-sp1)\*sp2\*(1-sp3) ##101

p1[7] <- pi1\*(1-se1)\*se2\*se3 + (1-pi1)\*sp1\*(1-sp2)\*(1-sp3) ###011

p1[8] <- pi1\*(1-se1)\*(1-se2)\*(1-se3) + (1-pi1)\*sp1\*sp2\*sp3 ###000

##population 2

p2[1] <- pi2\*se1\*se2\*se3 + (1-pi2)\*(1-sp1)\*(1-sp2)\*(1-sp3) ###111

p2[2] <- pi2\*se1\*(1-se2)\*(1-se3) + (1-pi2)\*(1-sp1)\*sp2\*sp3 ###100

p2[3] <- pi2\*(1-se1)\*se2\*(1-se3) + (1-pi2)\*sp1\*(1-sp2)\*sp3 ###010

p2[4] <- pi2\*(1-se1)\*(1-se2)\*se3 + (1-pi2)\*sp1\*sp2\*(1-sp3) ##001

p2[5] <- pi2\*se1\*se2\*(1-se3) + (1-pi2)\*(1-sp1)\*(1-sp2)\*sp3 ###110

p2[6] <- pi2\*se1\*(1-se2)\*se3 + (1-pi2)\*(1-sp1)\*sp2\*(1-sp3) ##101

p2[7] <- pi2\*(1-se1)\*se2\*se3 + (1-pi2)\*sp1\*(1-sp2)\*(1-sp3) ###011

p2[8] <- pi2\*(1-se1)\*(1-se2)\*(1-se3) + (1-pi2)\*sp1\*sp2\*sp3 ###000

###likelihood of contingency tables

t1[1:8] ~ dmulti(p1[1:8], n1)

t2[1:8] ~ dmulti(p2[1:8], n2)

# Prior part:

pi1 ~ dbeta(2, 2)

pi2 ~ dbeta(2, 8)

se1 ~ dbeta(1, 1)

se2 ~ dbeta(1, 1)

se3 ~ dbeta(2, 2)

sp1 ~ dbeta(90, 10)

sp2 ~ dbeta(90, 10)

sp3 ~ dbeta(90, 10)

# Hooks for automatic integration with R:

#data# t1, t2, n1, n2

#monitor# pi1, pi2, se1, se2, se3, sp1, sp2, sp3

#inits# se2

}

1. Provide the data of the cell counts

t1 = c(0, 3, 6, 2, 0, 1, 7, 131)

n1=150

t2 = c(3, 8, 3, 11, 0, 0, 11, 164)

n2=200

1. Provide initial values and run the model

se2=list(chain1=0.2, chain2=0.8)

results <- run.jags('3\_test\_mod.bug', adapt = 1000, n.chains=2,

burnin=1000, sample=10000)

plot(results, vars = "se1")

plot(results, vars = "se2")

plot(results, vars = "se3")

plot(results, vars = "pi1")

plot(results, vars = "pi2")

plot(results, vars = "sp1")

plot(results, vars = "sp2")

plot(results, vars = "sp3")

results

**Exercise 5 – three tests with conditional dependence**

Now we'll assume that tests 2 and 3 are conditionally dependent (for example the measure the same "thing/antibodies") so we expect the test from the first test to carry information on the result to the second test. We need to take into account this dependency by including parameters/terms that capture this conditional dependence.

1. Load the dataset that is as a csv file (ex45\_data.csv) in your working directory

df <- read.csv("ex45\_data.csv")

install.packages("dplyr")

library(dplyr)

1. Create a contingency table for each population

###contingency tables

df %>% mutate(x111 = ifelse(test1==1 & test2==1 & test3==1, 1, 0),

x100 = ifelse(test1==1 & test2==0 & test3==0, 1, 0),

x010 = ifelse(test1==0 & test2==1 & test3==0, 1, 0),

x001 = ifelse(test1==0 & test2==0 & test3==1, 1, 0),

x110 = ifelse(test1==1 & test2==1 & test3==0, 1, 0),

x101 = ifelse(test1==1 & test2==0 & test3==1, 1, 0),

x011 = ifelse(test1==0 & test2==1 & test3==1, 1, 0),

x000 = ifelse(test1==0 & test2==0 & test3 == 0, 1, 0)) %>%

group\_by(Population) %>%

summarise(x111=sum(x111), x100=sum(x100), x010=sum(x010),

x001=sum(x001), x110=sum(x110), x101=sum(x101),

x011=sum(x011), x000=sum(x000))

1. Write the model and save as “3\_test\_cov.bug” three tests – cell probabilities:

model{

###population 1

p1[1] <- pi1\*se1\*(se2\*se3+CovSe23) + (1-pi1)\*(1-sp1)\*((1-sp2)\*(1-sp3)+CovSp23) ###111

p1[2] <- pi1\*se1\*((1-se2)\*(1-se3)+CovSe23) + (1-pi1)\*(1-sp1)\*(sp2\*sp3+CovSp23) ###100

p1[3] <- pi1\*(1-se1)\*(se2\*(1-se3)-CovSe23) + (1-pi1)\*sp1\*((1-sp2)\*sp3-CovSp23) ###010

p1[4] <- pi1\*(1-se1)\*((1-se2)\*se3-CovSe23) + (1-pi1)\*sp1\*(sp2\*(1-sp3)-CovSp23) ##001

p1[5] <- pi1\*se1\*(se2\*(1-se3)-CovSe23) + (1-pi1)\*(1-sp1)\*((1-sp2)\*sp3-CovSp23) ###110

p1[6] <- pi1\*se1\*((1-se2)\*se3-CovSe23) + (1-pi1)\*(1-sp1)\*(sp2\*(1-sp3)-CovSp23) ##101

p1[7] <- pi1\*(1-se1)\*(se2\*se3+CovSe23) + (1-pi1)\*sp1\*((1-sp2)\*(1-sp3)+CovSp23) ###011

p1[8] <- pi1\*(1-se1)\*((1-se2)\*(1-se3)+CovSe23) + (1-pi1)\*sp1\*(sp2\*sp3+CovSp23) ###000

##population 2

p2[1] <- pi2\*se1\*(se2\*se3+CovSe23) + (1-pi2)\*(1-sp1)\*((1-sp2)\*(1-sp3)+CovSp23) ###111

p2[2] <- pi2\*se1\*((1-se2)\*(1-se3)+CovSe23) + (1-pi2)\*(1-sp1)\*(sp2\*sp3+CovSp23) ###100

p2[3] <- pi2\*(1-se1)\*(se2\*(1-se3)-CovSe23) + (1-pi2)\*sp1\*((1-sp2)\*sp3-CovSp23) ###010

p2[4] <- pi2\*(1-se1)\*((1-se2)\*se3-CovSe23) + (1-pi2)\*sp1\*(sp2\*(1-sp3)-CovSp23) ##001

p2[5] <- pi2\*se1\*(se2\*(1-se3)-CovSe23) + (1-pi2)\*(1-sp1)\*((1-sp2)\*sp3-CovSp23) ###110

p2[6] <- pi2\*se1\*((1-se2)\*se3-CovSe23) + (1-pi2)\*(1-sp1)\*(sp2\*(1-sp3)-CovSp23) ##101

p2[7] <- pi2\*(1-se1)\*(se2\*se3+CovSe23) + (1-pi2)\*sp1\*((1-sp2)\*(1-sp3)+CovSp23) ###011

p2[8] <- pi2\*(1-se1)\*((1-se2)\*(1-se3)+CovSe23) + (1-pi2)\*sp1\*(sp2\*sp3+CovSp23) ###000

###likelihood of contingency tables

t1[1:8] ~ dmulti(p1[1:8], n1)

t2[1:8] ~ dmulti(p2[1:8], n2)

# Prior part:

pi1 ~ dbeta(1, 3)

pi2 ~ dbeta(2, 2)

se1 ~ dbeta(2, 2)

se2 ~ dbeta(2, 2)

se3 ~ dbeta(2, 2)

sp1 ~ dbeta(99, 1)

sp2 ~ dbeta(95, 5)

sp3 ~ dbeta(95, 5)

alpha.CovSe23 <- max(-(1-se2)\*(1-se3), -se2\*se3)

beta.CovSe23 <- min(se2\*(1-se3), se3\*(1-se2))

gamma.CovSp23 <- max(-(1-sp2)\*(1-sp3), -sp2\*sp3)

delta.CovSp23 <- min(sp2\*(1-sp3), sp3\*(1-sp2))

CovSe23 ~ dunif(alpha.CovSe23, beta.CovSe23)

CovSp23 ~ dunif(gamma.CovSp23, delta.CovSp23)

# Hooks for automatic integration with R:

#data# t1, t2, n1, n2

#monitor# pi1, pi2, se1, se2, se3, sp1, sp2, sp3, CovSe23, CovSp23

#inits# se2

}

1. Provide the data of the cell counts

t1 = c(0, 3, 6, 2, 0, 1, 7, 131)

n1=150

t2 = c(3, 8, 3, 11, 0, 0, 11, 164)

n2=200

1. Provide initial values and run the model

se2=list(chain1=0.2, chain2=0.8)

results <- run.jags('3\_test\_cov.bug', adapt = 1000, n.chains=2,

burnin=1000, sample=10000)

plot(results, vars = "se1")

plot(results, vars = "se2")

plot(results, vars = "se3")

plot(results, vars = "pi1")

plot(results, vars = "pi2")

plot(results, vars = "sp1")

plot(results, vars = "sp2")

plot(results, vars = "sp3")

plot(results, vars = "CovSe23")

plot(results, vars = "CovSp23")

results