

# Measures of accuracy, binary diagnostic tests

*Saturday, Nov 07, 2015*

- 1 Introduction
- 2 Basic measures of diagnostic accuracy are calculated and presented using frequentist and Bayesian approaches.
- 3 Note in the example below, ‘ppv’ and ‘npv’ are estimated from the data. The Bayesian ‘ppv2’ and ‘npv2’ estimates are based on a user defined prevalence designed to mimic the observed prevalence.
- 4 When calculating positive predictive value and negative predictive value measures for a user defined prevalence, the frequentist approach employed here uses a prior point estimate whilst the Bayesian approach uses a prior distribution. The user defined prevalence point estimate is ‘prev.2’ and the respective conditional probabilities ‘ppv3’ and ‘npv3’. Compare the confidence intervals provided by the frequentist and Bayesian approaches.
- 5 “In an effort to introduce the least possible amount of external information into a Bayesian analysis, improper densities sometimes are used as priors. This must be done with great care. Using a proper prior guarantees that the posterior also will be proper, but an improper prior may produce an improper posterior. If the posterior density is improper, it doesn’t exist, so no valid inference can come out of it. Thus, if you choose to use an improper prior, you must verify that the resulting posterior is proper” p73, also see page 78 ‘Applied Bayesian Statistics With R and OpenBUGS Examples’.

- 6 “An improper prior implies that the Bayesian and conventional analyses will provide very similar estimates and standard errors. Note that an improper prior can be used if there are no cells with a zero count, and if some of the cells have zero counts, a uniform prior may be used when very little prior information is available.” p180 Advanced Bayesian Methods for Medical Test Accuracy. “... because it would result in an improper posterior density when the cell frequencies are zero. For ‘small’  $p$  ( $=0.01$  and  $0.05$ ), some of the cell frequencies are in fact zero, thus, a uniform prior was employed instead...” p324.
- 7 “... When the sample size is ‘small’ the posterior analysis with a uniform prior will differ from that with an improper prior...” p61 Bayesian Methods in Epidemiology, see p65 section 2.5 also...
- 8 If there are zeros in the data the improper prior Bayesian model below will fail and the program will not complete.
- 9 Note, OpenBUGS software needs to be installed on your computer.

## 10 Definitions

- 10.1 Sensitivity is the proportion of true positives that are correctly identified by the test.
- 10.2 Specificity is the proportion of true negatives that are correctly identified by the test.
- 10.3 The positive predictive value (PPV) is the proportion of patients with positive test results who are correctly diagnosed. (Given a positive test, the PPV is the probability of disease).
- 10.4 The negative predictive value (NPV) is the proportion of patients with negative test results who are correctly diagnosed. (Given a negative test, the NPV is the probability of no disease)
- 10.5 The likelihood ratio for a positive result tells you how much the odds of the disease increase when a test is positive.
- 10.6 The likelihood ratio for a negative result tells you how much the odds of the disease decrease when a test is negative.
- 10.7 The prevalence is the unconditional probability of disease in the population of interest. That is the probability that a randomly chosen individual from the population of interest is diseased. It is informative to compare the prevalence with the PPV and 1-prevalence with the NPV to see how the probabilities change before and after diagnostic testing. The likelihood ratios are independent of prevalence. PPV and NPV are dependent on the prevalence.

## 11 Set up Rmarkdown environment

```
rm(list=ls())
startTime<-proc.time()
library(knitr)
options(width=120)
opts_chunk$set(comment = "", warning = FALSE, message = FALSE,
              echo = FALSE, tidy = FALSE, size="tiny", cache=FALSE,
              progress=TRUE,
              cache.path = 'program_Cache/',
              fig.path='figure/')

knitr:::knit_hooks$set(inline = function(x) {
knitr:::format_sci(x, 'md')
})
```

## 12 R packages and rounding functions

```
list.of.packages <- c("binom" , "Hmisc", "epitoools", "R2OpenBUGS", "knitr", "xtable", "LearnBayes")

new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[, "Package"])] 

if(length(new.packages)) install.packages(new.packages)

sapply(X = list.of.packages, require, character.only = TRUE)

#rounding functions

p1x <- function(x) {print(formatC(x, format="f", digits=1), quote=FALSE)}
p2x <- function(x) {print(formatC(x, format="f", digits=2), quote=FALSE)}
p3x <- function(x) {print(formatC(x, format="f", digits=3), quote=FALSE)}
p4x <- function(x) {print(formatC(x, format="f", digits=4), quote=FALSE)}
p5x <- function(x) {print(formatC(x, format="f", digits=5), quote=FALSE)}

#not used: but perhaps help colour plot text based on loop count

is.even <- function(x){ x %% 2 == 0 }
```

## 13 Data, a number of different data sets are presented.

```
set.seed(123)    # Reproducible results
n.sims <- 10000 # No of Monte Carlo simulations for frequentist confidence intervals
prev.2 <- 0.2    # User defined prevalence, see estimates of PPV and NPV: 'ppv3' and 'npv3'

# 157/308 disease free and 32/52 diseased have positive test results

#      a00=151
#      a01=20
#      a10=157
#      a11=32

# 95/156 disease free and 27/35 diseased have positive test results

#      a00=61
#      a01=8
#      a10=95
#      a11=27

# Bayesian Methods in Epidemiology p284
# Advanced Bayesian Methods for Medical Test Accuracy p51
# 115/442 disease free and 818/1026 diseased have positive test results

#      a00=327
#      a01=208
#      a10=115
#      a11=818

# STATISTICS WITH CONFIDENCE Altman 200 p109...
# 16/80 disease free and 36/40 diseased have positive test results

#      a00=64
#      a01=4
#      a10=16
#      a11=36

# Bayesian Methods in Epidemiology table 7.25
# 2908/73113 disease free and 262/287 diseased have positive test results

#      a00=70205
#      a01=25
#      a10=2908
#      a11=262

# Bayesian Methods in Epidemiology table 7.26
# 3216/70332 disease free and 76/117 diseased have positive test results

#      a00=67116
#      a01=41
#      a10=3216
#      a11=76
```

```

# Make up some data with low count in cells to see performance

#      a00=64
#      a01=1
#      a10=16
#      a11=36

# breast mammogram data,    http://theincidentaleconomist.com/wordpress/healthcare-triage-bayes-theor

#      a00=127344
#      a01=118
#      a10=13212
#      a11=610

# http://jco.ascopubs.org/content/29/35/4620.full.pdf
# Development and Independent Validation of a Prognostic Assay for Stage II Colon Cancer Using Form

N1<-59 # true positives
S1<-33 # positive calls
N2<-85 # true negatives
S2<-61 # negative calls

a00=S2
a01=N1-S1
a10=N2-S2
a11=S1

```

## 14 Contingency Table

```

t2 <- matrix(c(a00,a10,a01,a11 ),ncol=2,byrow=FALSE)
colnames(t2) <- c("No Dis","Dis")
rownames(t2) <- c("-ve","+ve")

t2 <- as.table(t2)

df <- expand.table(t2)
tb <- with(df,table(Var2, Var1 ))
dd <- addmargins(tb, FUN = list>Total = sum, quiet = TRUE)

kable(dd, digits=2)

```

	No Dis	Dis	Total
-ve	61	26	87
+ve	24	33	57
Total	85	59	144

## 15 Frequentist: Measures of Diagnostic Accuracy

### 15.1 Population prevalence estimate from the sample

```
method x n      mean      lower      upper
1 wilson 59 144 0.4097222 0.3327607 0.4913752
```

### 15.2 Sensitivity and CI

```
method x n      mean      lower      upper
1 wilson 33 59 0.559322 0.4328935 0.6784979
```

### 15.3 Specificity and CI

```
method x n      mean      lower      upper
1 wilson 61 85 0.7176471 0.6141608 0.8023115
```

### 15.4 PPV and CI based on prevalence in data

```
method x n      mean      lower      upper
1 wilson 33 57 0.5789474 0.4498014 0.698124
```

### 15.5 NPV and CI based on prevalence in data

```
method x n      mean      lower      upper
1 wilson 61 87 0.7011494 0.5981275 0.7871591
```

### 15.6 Positive likelihood ratio; ‘probability of positive test in those with disease’/‘probability of positive test in those without disease’

```
sens <- (a11)/(a11+a01)
spec <- (a00)/(a00+a10)

p4x(sens/(1-spec))
```

```
[1] 1.9809
```

### 15.7 Negative likelihood ratio; ‘probability of negative test in those with disease’/‘probability of negative test in those without disease’

```
p4x((1-sens)/(spec))
```

```
[1] 0.6141
```

- 15.8 Positive likelihood ratio alternative approach, the odds of disease post positive test, divided by the odds of disease prior to testing.

```
(pre.test.odds<-(a01+a11 )/ (a00+a10 )) #marginal
```

```
[1] 0.6941176
```

```
(post.test.odds<-(a11)/(a10)) #given pos result
```

```
[1] 1.375
```

```
(LRpos<-post.test.odds/pre.test.odds)
```

```
[1] 1.980932
```

- 15.9 The test is positive about 1.9809322 times more often among the diseased, compared to those without disease.

- 15.10 Negative likelihood ratio alternative approach, that is, the odds of disease post negative test, divided by those prior to testing. It is important to note that likelihood ratios always refer to the likelihood of having disease; the positive and negative designation simply refers to the test result. Hence the interpretation of the post-test odds is always a likelihood of having disease.

```
(post.test.odds<-(a01)/(a00)) #given neg result
```

```
[1] 0.4262295
```

```
(LRneg<- post.test.odds/pre.test.odds)
```

```
[1] 0.6140595
```

- 15.11 On the other hand among those who have the disease, the test is negative 0.6140595 times less often compared to those without the disease.

## 15.12 Confidence intervals for likelihood ratios (care required for se calculation), positive

```
(dlr<-sens/(1-spec))
```

```
[1] 1.980932
```

```
(dlr<-((a11)/(a11+a01)) / ((a10)/(a00+a10)))
```

```
[1] 1.980932
```

```
one <- 1/a11
two <- 1/(a11+a01)
three <-1/a10
four <- 1/(a00+a10)

log.se<- sqrt(one - two + three - four)
(plr.ci<-exp(log(dlr)+(c(-1,1)*(1.96*(log.se)))))
```

```
[1] 1.317750 2.977872
```

## 15.13 Negative likelihood ratio 95% confidence interval

```
(dlr<-(1-sens)/(spec))
```

```
[1] 0.6140595
```

```
(dlr<-((a01)/(a11+a01)) / ((a00)/(a00+a10)))
```

```
[1] 0.6140595
```

```
one <- 1/a01
two <- 1/(a11+a01)
three <-1/a00
four <- 1/(a00+a10)

log.se<- sqrt(one - two + three - four)
(nlr.ci<-exp(log(dlr)+(c(-1,1)*(1.96*(log.se)))))
```

```
[1] 0.4472844 0.8430184
```

### 15.14 Predicted values based on user defined prevalence

```
#prev<-(a11+a01) / a00+a01+a10+a11 #estimate of prevalence from sample  
prev<-prev.2  
false_neg <- 1-sens  
false_pos <- 1-spec
```

### 15.15 Positive predicted value point estimate based on user defined prevalence of 0.2

```
(f.ppv2<-(sens*prev) / ((sens*prev)+(false_pos*(1-prev))))
```

[1] 0.3312079

### 15.16 Negative predicted value point estimate based on user defined prevalence of 0.2

```
(f.npv2<- (spec*(1-prev) / ((spec*(1-prev))+(false_neg*prev))))
```

[1] 0.8669156

### 15.17 Obtain confidence intervals for PPV and NPV. Simulate proportion of positives using observed sensitivity and diseased sample. Similarly simulate proportion of negatives using observed specificity and non diseased sample.

```
m1 <- rbinom (n.sims, (a01+a11), (a11) / (a01+a11) )  
m2 <- rbinom (n.sims, (a00+a10), (a00) / (a00+a10) )
```

### 15.18 For each simulation generate a PPV and NPV and obtain 0.025 and 0.975 percentiles

```
sens <- m1/(a11+a01)  
spec <- m2/(a00+a10)  
false_neg <- 1-sens  
false_pos <- 1-spec  
p1<-(sens*prev)/((sens*prev)+(false_pos*(1-prev)))  
n1<-(spec*(1-prev))/((spec*(1-prev))+(false_neg*prev))
```

### 15.19 PPV 0.025 and 0.975 percentiles based on user defined prevalence of 0.2

```
(f.ppv.ci<- (quantile (p1, c(.025, .975))))
```

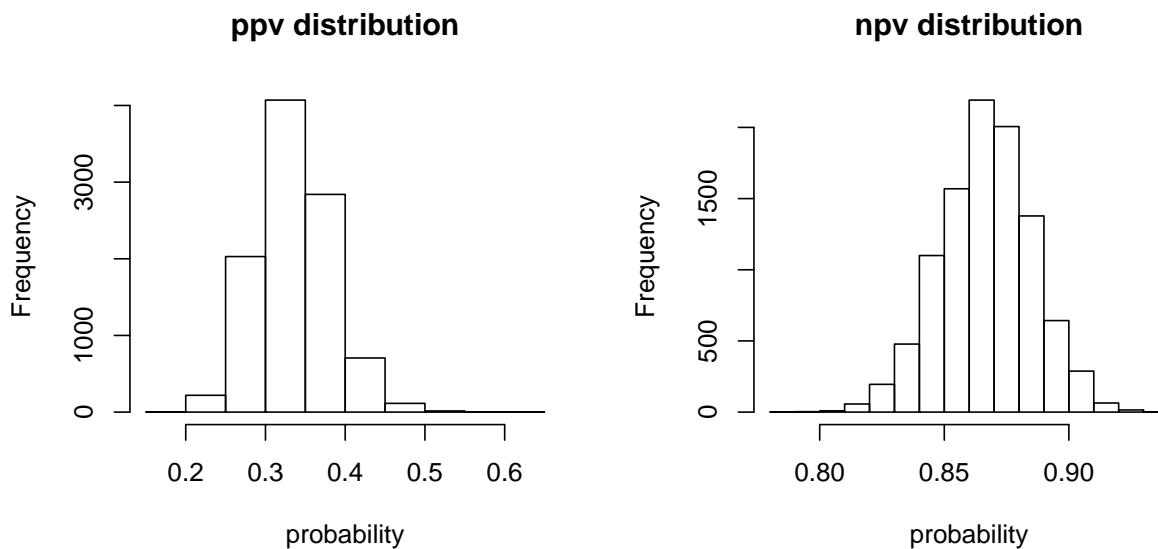
```
2.5%      97.5%
0.2511216 0.4335434
```

## 15.20 NPV 0.025 and 0.975 percentiles based on user defined prevalence of 0.2

```
(f.npv.ci<- (quantile (n1, c(.025, .975))))
```

```
2.5%      97.5%
0.8293173 0.9029237
```

```
par(mfrow=c(1,2))
hist(p1, main="ppv distribution", xlab="probability")
hist(n1, main="npv distribution", xlab="probability")
```



```
par(mfrow=c(1,1))
```

## 16 Bayesian: Measures of Diagnostic Accuracy

- 16.1 For the Bayesian example, ‘ppv’ and ‘npv’ are estimated from the data. ‘ppv2’ and ‘npv2’ are alternative estimates of the same estimands, but using a beta prior ‘beta(a, b)’ for the prevalence distribution which attempts to mimic the population prevalence calculated from the sample. ‘ppv3’ and ‘npv3’ are further alternative estimates, but this time based on a sample size of 100 and using a user defined prevalence distribution ‘beta(a2, b2)’ distinct from that observed in the sample. (Note, ‘ppv2’ and ‘npv2’ are only to be found in the Bayesian analysis.)

```
(foo<-binom.confint( a11+a01, a00+a01+a10+a11 ,method="wilson"))
```

```
method  x    n      mean      lower      upper
1 wilson 59 144 0.4097222 0.3327607 0.4913752
```

```
#      coded out as sometimes the beta.select function throws an error
#
#      quantile1=list(p=.025, x=foo$lower)      # the 2.5% quantile
#      quantile2=list(p=.975, x=foo$upper)      # the 97.5% quantile
#      a<-beta.select(quantile1, quantile2)[1]
#      b<-beta.select(quantile1, quantile2)[2]
#      qbeta(c(.025, .975), a, b)
#      foo

a<-a11+a01  ## idea here is use the prevalence in the observed sample, a/(a+b)
b<-a00+a10  ## remember, the mean of beta dist is a/(a+b)
qbeta(c(.025, .975), a, b)
```

```
[1] 0.3309899 0.4908304
```

```
a2<- prev.2*100  ##a defined prevalence as a fraction, so N=100
b2<- 100-a2
qbeta(c(.025, .975), a2, b2)
```

```
[1] 0.1279847 0.2833676
```

## 16.2 Plot beta distribution(s) (code from R documentation help file)

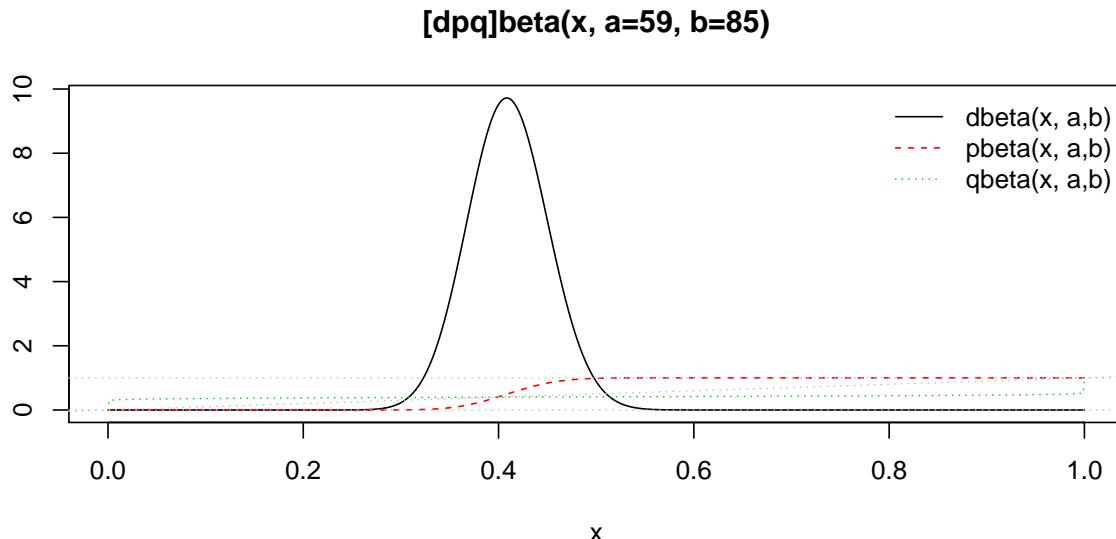
```

pl.beta <- function(a,b, asp = if(isLim) 1, ylim = if(isLim) c(0,1.1)) {
  if(isLim <- a == 0 || b == 0 || a == Inf || b == Inf) {
    eps <- 1e-10
    x <- c(0, eps, (1:7)/16, 1/2+c(-eps,0,eps), (9:15)/16, 1-eps, 1)
  } else {
    x <- seq(0, 1, length = 1025)
  }
  fx <- cbind(dbeta(x, a,b), pbeta(x, a,b), qbeta(x, a,b))
  f <- fx; f[fx == Inf] <- 1e100

  matplot(x, f, ylab="", type="l", ylim=ylim, asp=asp,
           main = sprintf("[dpq]beta(x, a=%g, b=%g)", a,b))
  abline(0,1, col="gray", lty=3)
  abline(h = 0:1, col="gray", lty=3)
  legend("topright", paste0(c("d","p","q"), "beta(x, a,b)"),
         col=1:3, lty=1:3, bty = "n")
  invisible(cbind(x, fx))
}

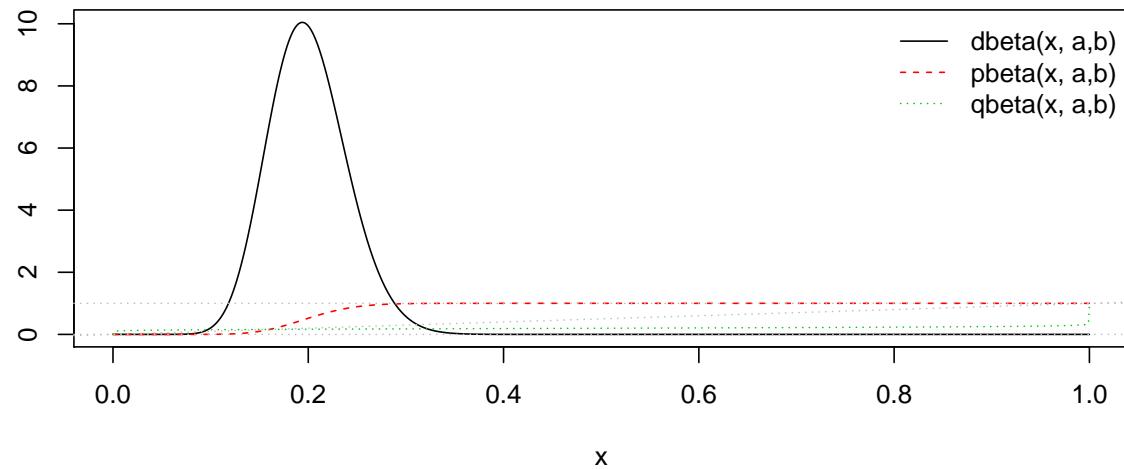
```

pl.beta(a,b)



pl.beta(a2,b2)

[dpq]beta(x, a=20, b=80)



### 16.3 Bayesian Model (see Lyle D. Broemeling references)

```
cat("model{  
  
    # Dirichlet distribution for cell probabilities  
  
    g00~dgamma(a00,2)  
    g01~dgamma(a01,2)  
    g10~dgamma(a10,2)  
    g11~dgamma(a11,2)  
  
    h<-g00+g01+g10+g11  
  
    # the theta have a Dirichlet distribution  
  
    theta00<-g00/h  
    theta01<-g01/h  
    theta10<-g10/h  
    theta11<-g11/h  
  
    # calculation of basic test accuracy statistics  
  
    tpf<-theta11/(theta11+theta01)  
    se<-tpf  
    sp<-1-fpf  
    fpf<-theta10/(theta10+theta00)  
    tnf<-theta00/(theta00+theta10)  
    fnf<-theta01/(theta01+theta11)  
    ppv<-theta11/(theta10+theta11)  
    npv<-theta00/(theta00+theta01)  
    pdlr<-tpf/fpf  
    ndlr<-fnf/tnf  
  
    # user defined prevalence  
    # p is a distribution of a prevalence of interest rather than a point estimate  
  
    p ~ dbeta(a,b)  
  
    false_neg <- 1-se  
    false_pos <- 1-sp  
  
    ppv2<- ( se*p ) / ( (se*p) + (false_pos*(1-p) ) )  
    npv2<- ( sp*(1-p) ) / ( (sp*(1-p)) + (false_neg*p) ) )  
  
    ### another prevalence distribution to estimate the predictive values  
  
    p3 ~ dbeta(a2,b2)  
  
    ppv3<- ( se*p3 ) / ( (se*p3) + (false_pos*(1-p3) ) )  
    npv3<- ( sp*(1-p3) ) / ( (sp*(1-p3)) + (false_neg*p3) ) )  
  
} ", file="model.txt")
```

16.4 The data. By adding a one to each cell, one is in effect assuming a uniform prior for the cell probabilities. The data includes objects ‘a’ and ‘b’ for a prevalence distribution so that PPV and NPV can be estimated assuming a different prevalence. Included also are ‘a2’ and ‘b2’ for a prevalence distribution, so that PPV and NPV can be estimated assuming a different prevalence.

```
ad<-0 #improper ad=0, uniform (proper) ad=1  
data<- list( a00=a00+ad, a01=a01+ad, a10=a10+ad, a11=a11+ad, a=a, b=b ,a2=a2, b2=b2 )
```

16.5 Initial values for MCMC and number of chains

```
chains=3  
u1<-1.0  
u2<-0.5  
u3<-0.1  
  
user.initial <- list(  
  
  list( g00=u1,g01=u1,g10=u1,g11=u1),  
  
  list( g00=u2,g01=u2,g10=u2,g11=u2),  
  
  list( g00=u3,g01=u3,g10=u3,g11=u3)  
)
```

16.6 Parameters to monitor and collect

```
parz = c("se","sp","tpf","fpf","ppv","npv","ppv2","npv2","pdlr","ndlr","ppv3","npv3")
```

16.7 Execute analysis, supply iterations and burn in...

```
T <- 55000  
B <- 5000  
  
res <- bugs(data, inits=user.initial , parameters.to.save=parz,  
            model="model.txt", n.chains=chains,  
            n.iter=T, n.burnin=B, n.thin=1,  
            debug=F, DIC=FALSE, bugs.seed=2, codaPkg=F)
```

## 16.8 Re-execute analysis using different prior...

```
ad<-1  #improper ad=0, uniform ad=1
data<- list( a00=a00+ad, a01=a01+ad, a10=a10+ad, a11=a11+ad, a=a, b=b ,a2=a2, b2=b2 )

res1 <- bugs(data, inits=user.initial , parameters.to.save=parz,
              model="model.txt", n.chains=chains,
              n.iter=T, n.burnin=B, n.thin=1,
              debug=F, DIC=FALSE, bugs.seed=2, codaPkg=F)
```

## 16.9 Posterior estimates (proper prior)

```
print(res1, digits=5)
```

```
Inference for Bugs model at "model.txt",
Current: 3 chains, each with 55000 iterations (first 5000 discarded)
Cumulative: n.sims = 150000 iterations saved
      mean     sd   2.5%   25%   50%   75% 97.5%    Rhat  n.eff
se   0.55741 0.06316 0.4325 0.5146 0.5582 0.6007 0.6791 1.00103 45000
sp   0.71293 0.04820 0.6143 0.6811 0.7146 0.7463 0.8024 1.00100 150000
tpf  0.55741 0.06316 0.4325 0.5146 0.5582 0.6007 0.6791 1.00103 45000
fpf  0.28707 0.04820 0.1976 0.2537 0.2854 0.3189 0.3857 1.00100 140000
ppv  0.57648 0.06379 0.4496 0.5336 0.5771 0.6203 0.6989 1.00101 85000
npv  0.69676 0.04853 0.5979 0.6646 0.6983 0.7304 0.7873 1.00100 150000
ppv2 0.57410 0.06436 0.4461 0.5307 0.5749 0.6185 0.6973 1.00099 150000
npv2 0.69878 0.04879 0.5987 0.6666 0.7003 0.7329 0.7894 1.00102 60000
pdlr 1.99946 0.42259 1.3160 1.7020 1.9495 2.2410 2.9690 1.00101 90000
ndlr 0.62373 0.09928 0.4403 0.5549 0.6197 0.6880 0.8291 1.00102 57000
ppv3 0.32825 0.07036 0.2005 0.2787 0.3247 0.3743 0.4755 1.00101 110000
npv3 0.86508 0.03446 0.7896 0.8437 0.8680 0.8897 0.9236 1.00102 56000
```

For each parameter, n.eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

## 16.10 Posterior estimates (improper prior)

```
print(res ,digits=5)
```

```
Inference for Bugs model at "model.txt",
Current: 3 chains, each with 55000 iterations (first 5000 discarded)
Cumulative: n.sims = 150000 iterations saved
      mean     sd   2.5%   25%   50%   75% 97.5%    Rhat  n.eff
se   0.55943 0.06417 0.4324 0.5160 0.5603 0.6035 0.6830 1.00099 150000
sp   0.71800 0.04853 0.6186 0.6860 0.7198 0.7516 0.8081 1.00099 150000
tpf  0.55943 0.06417 0.4324 0.5160 0.5603 0.6035 0.6830 1.00099 150000
fpf  0.28200 0.04853 0.1919 0.2484 0.2802 0.3140 0.3814 1.00100 150000
ppv  0.57926 0.06494 0.4497 0.5357 0.5800 0.6238 0.7035 1.00100 150000
npv  0.70134 0.04875 0.6018 0.6690 0.7030 0.7352 0.7921 1.00101 110000
ppv2 0.57931 0.06503 0.4496 0.5356 0.5803 0.6241 0.7036 1.00101 110000
npv2 0.70129 0.04885 0.6013 0.6690 0.7028 0.7354 0.7922 1.00101 92000
pdlr 2.04589 0.44193 1.3330 1.7340 1.9940 2.2970 3.0630 1.00099 150000
ndlr 0.61650 0.09978 0.4321 0.5475 0.6123 0.6810 0.8239 1.00099 150000
ppv3 0.33302 0.07142 0.2033 0.2827 0.3296 0.3799 0.4819 1.00100 150000
npv3 0.86647 0.03435 0.7910 0.8451 0.8695 0.8911 0.9248 1.00100 150000
```

For each parameter, n.eff is a crude measure of effective sample size,  
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

16.11 Compare with frequentist estimates. Notice the ppv3 and npv3 confidence intervals are wider with the Bayesian approach, due to the fact the Bayesian prior prevalence is supplied as a distribution.

	2.5%	97.5%
se	0.55932	0.43289
sp	0.71765	0.61416
ppv	0.57895	0.44980
npv	0.70115	0.59813
LRpos	1.98093	1.31775
LRneg	0.61406	0.44728
ppv3	0.33121	0.25112
npv3	0.86692	0.82932
	0.43354	0.90292

16.12 Posterior density and MCMC chains, improper prior

```
J<- dimnames(res$sims.array)[[3]]
k<- length(J)

for (i in 1:k) {

  par(mfrow=c(1,2))

  f<-as.vector(res$sims.array[, , J[i]])

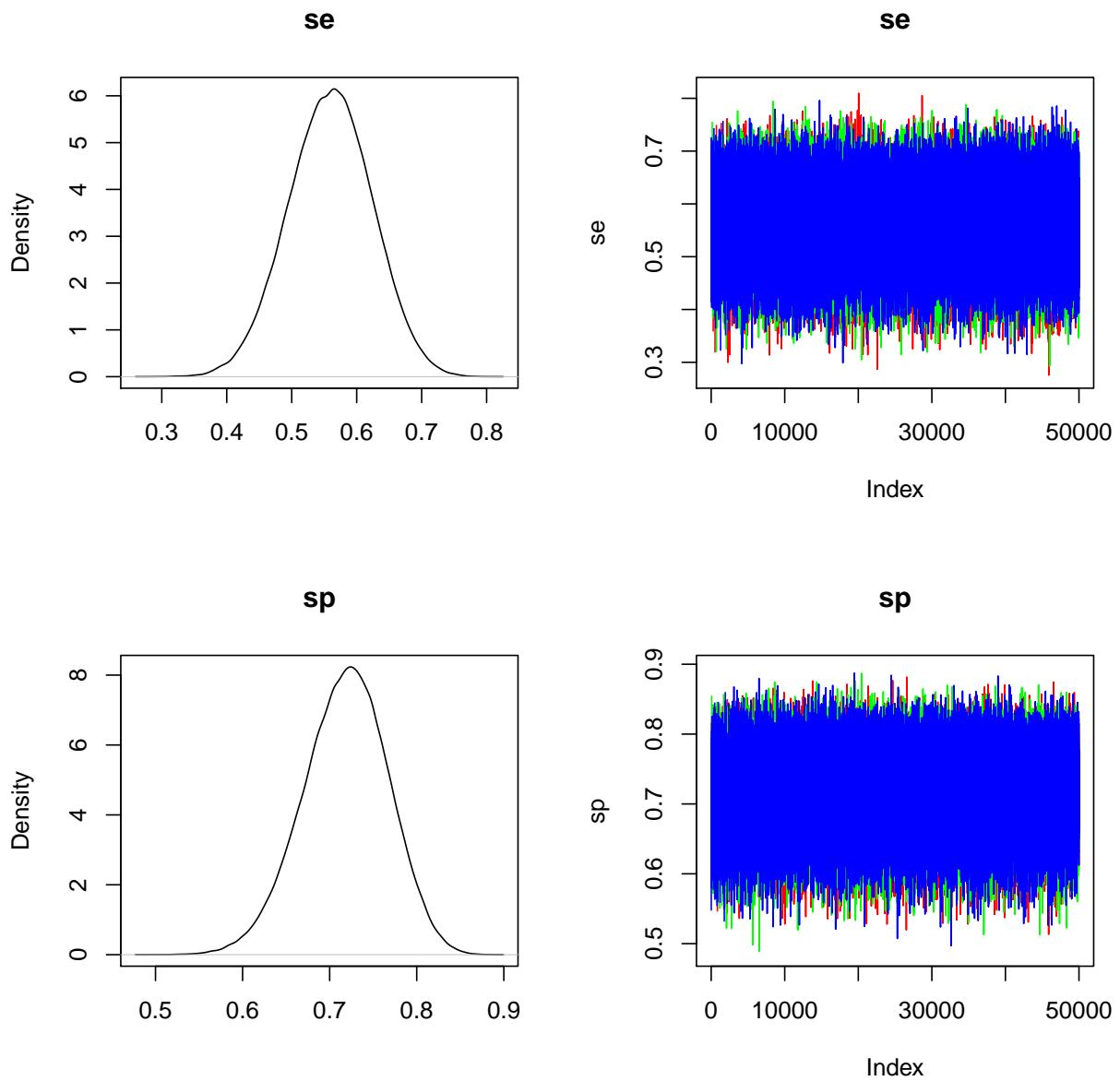
  plot(density(f), main=J[i], xlab="")

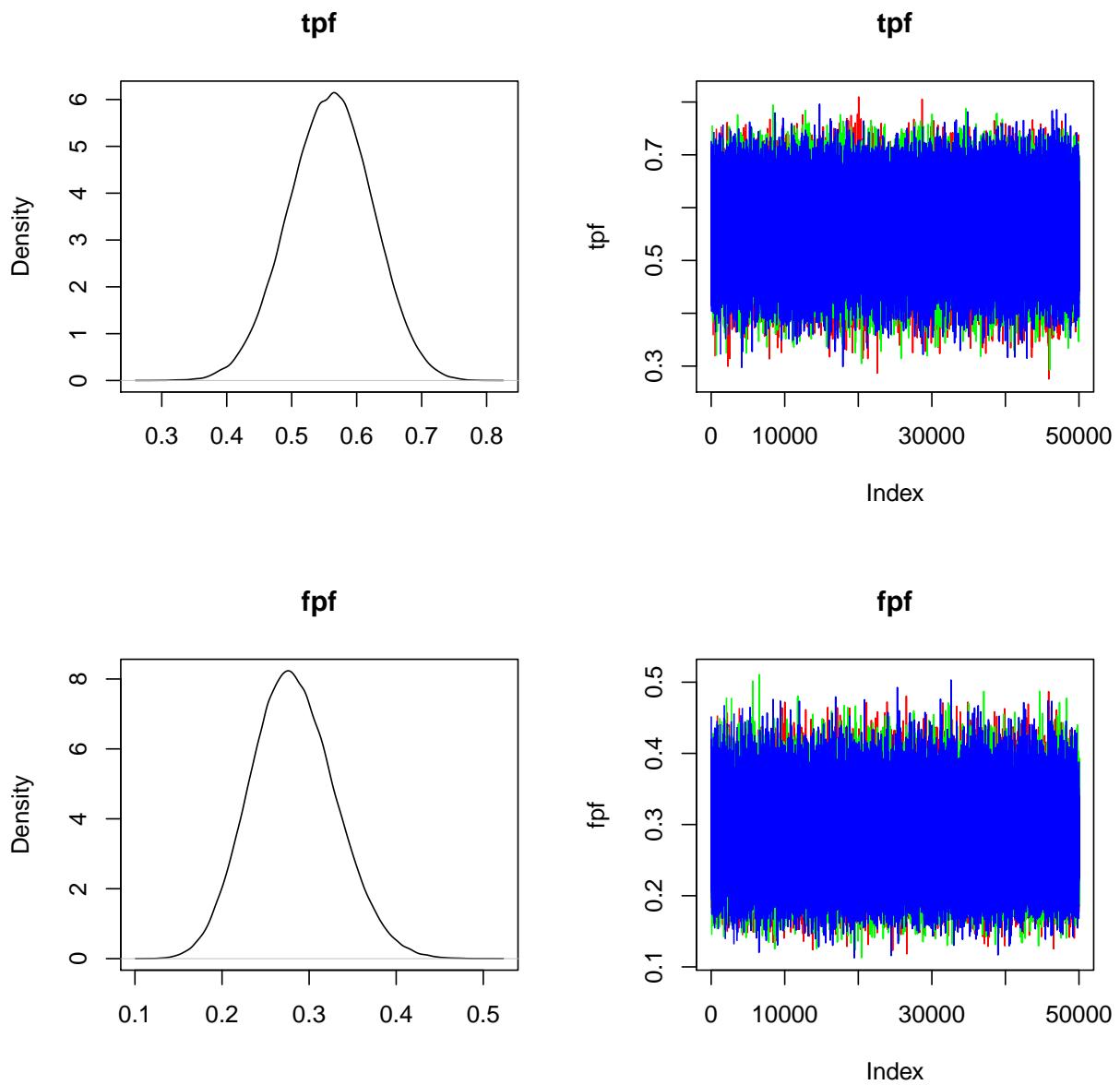
  x<-rainbow(chains)

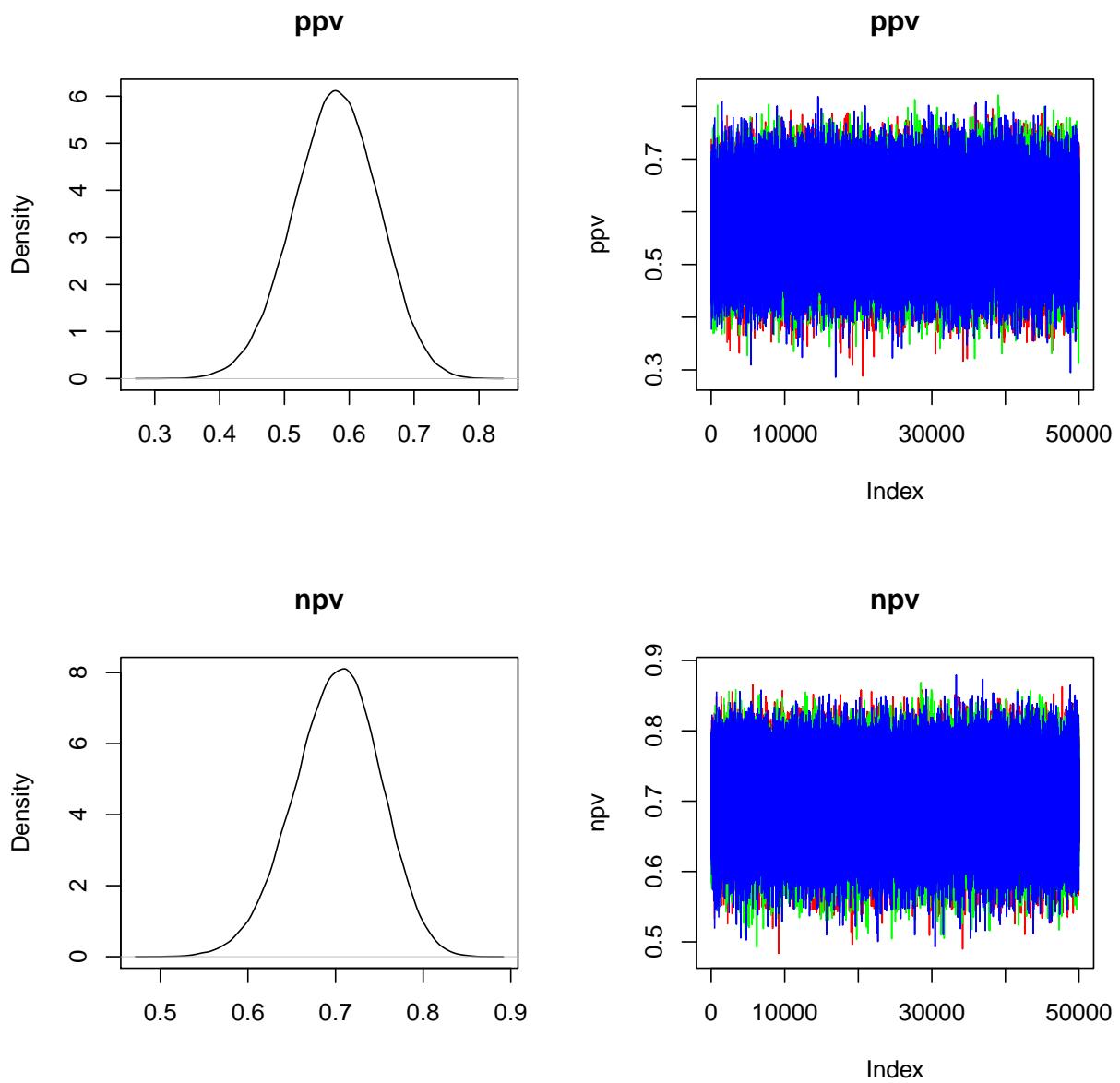
  plot(res$sims.array[,1, i], col=x[1], type = "l", main=J[i],
       ylim=c(min(f)*.99,max(f)*1.01), ylab=J[i])

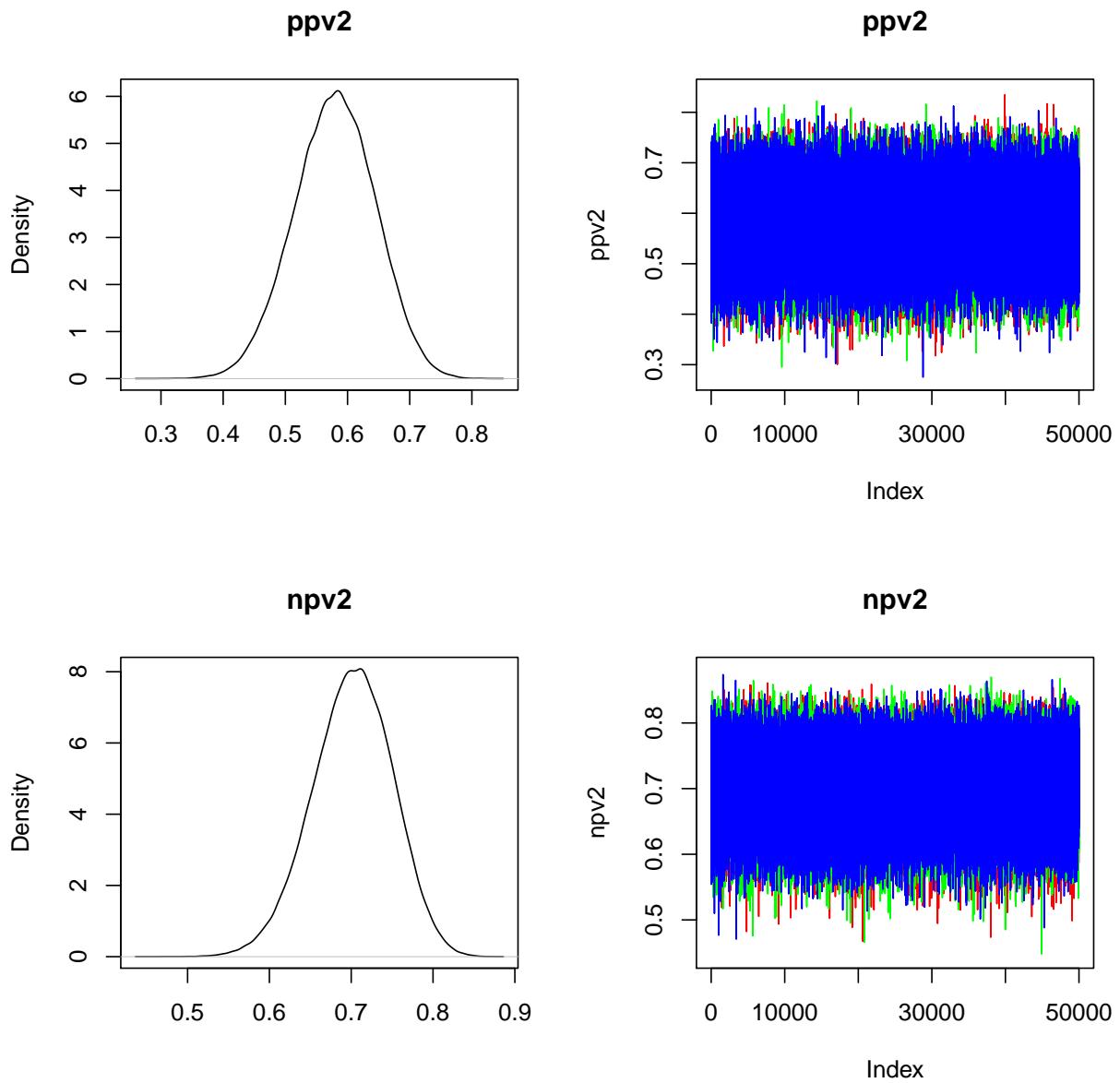
  ## add the remaining chains
  for (ch in 2:(chains)) {
    lines(res$sims.array[,ch, i], col=x[ch] )
  }

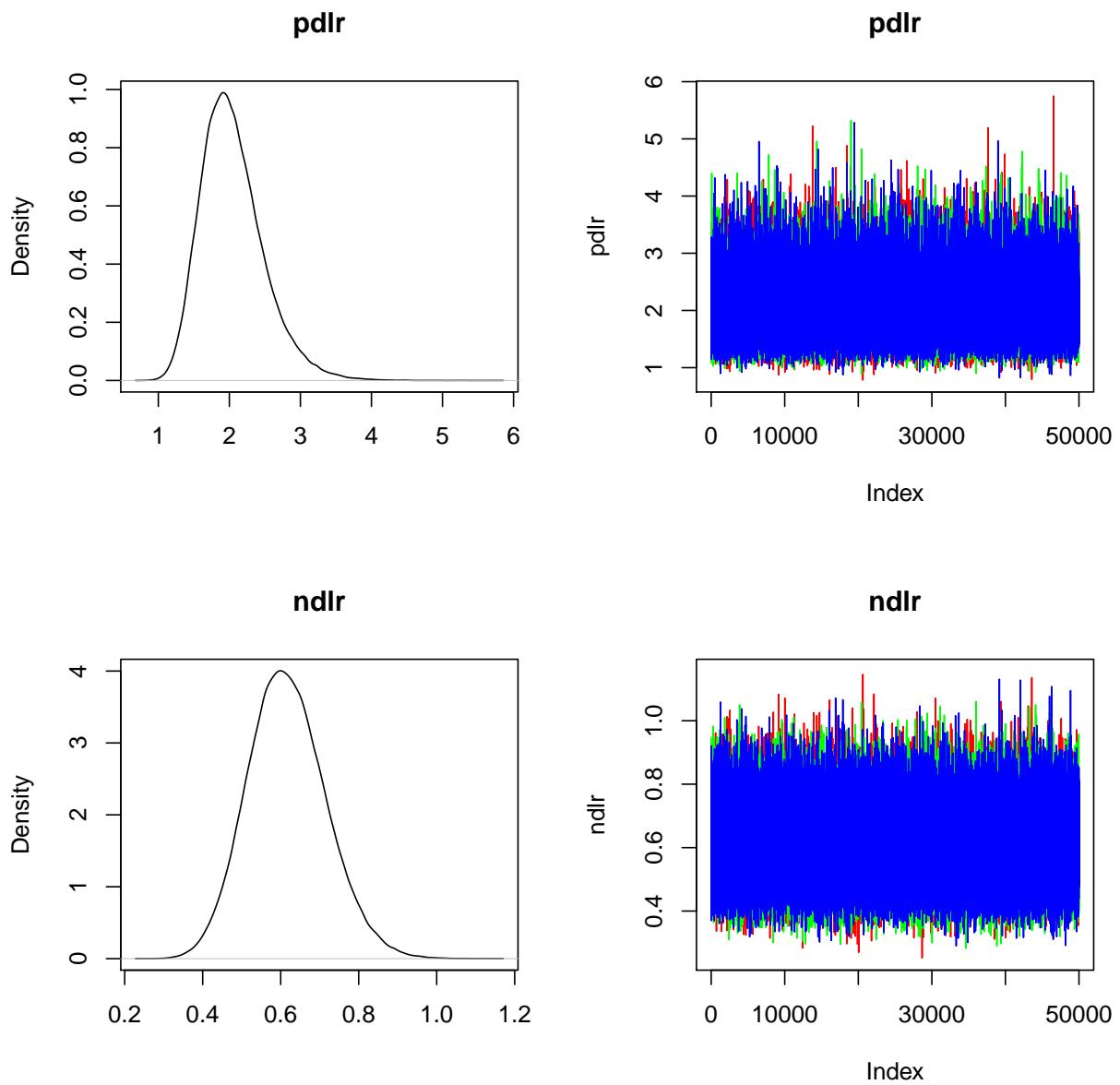
  par(mfrow=c(1,1))
}
```

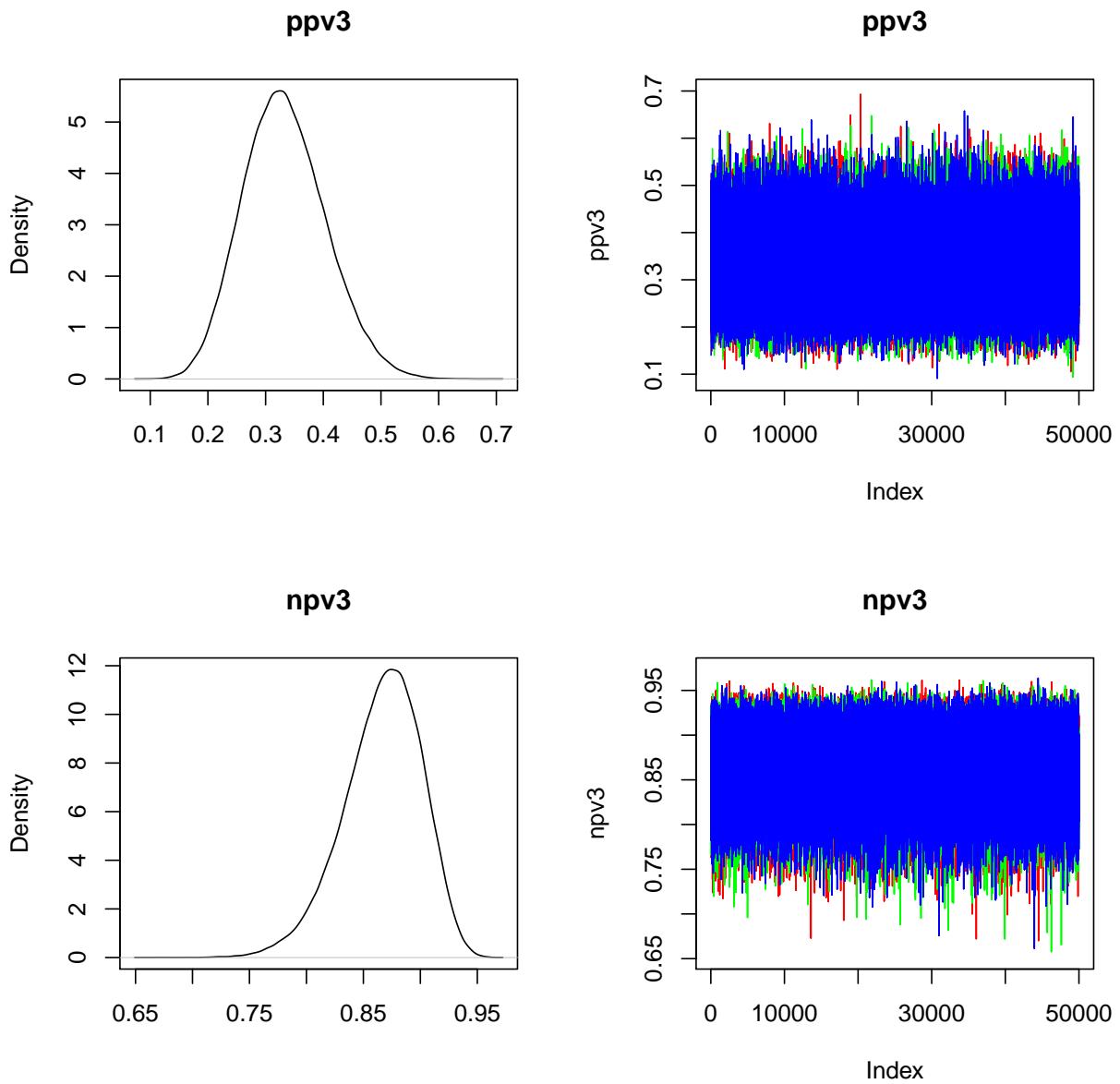












### 16.13 Posterior density and MCMC chains, proper prior

```

J<- dimnames(res1$sims.array)[[3]]
k<- length(J)

for (i in 1:k) {

  par(mfrow=c(1,2))

  f<-as.vector(res1$sims.array[, , J[i]])

  plot(density(f), main=J[i], xlab="")
}

```

```

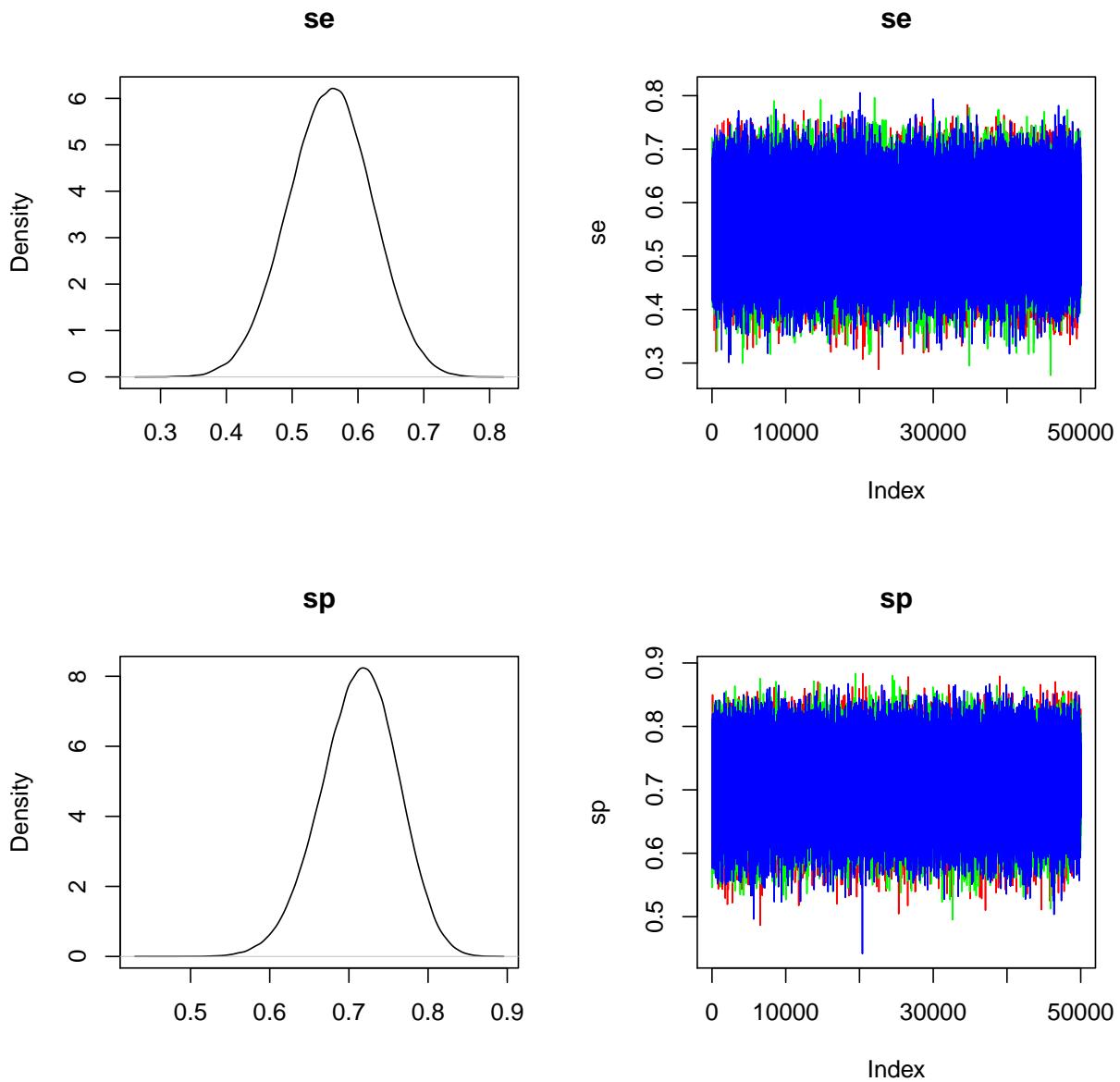
x<-rainbow(chains)

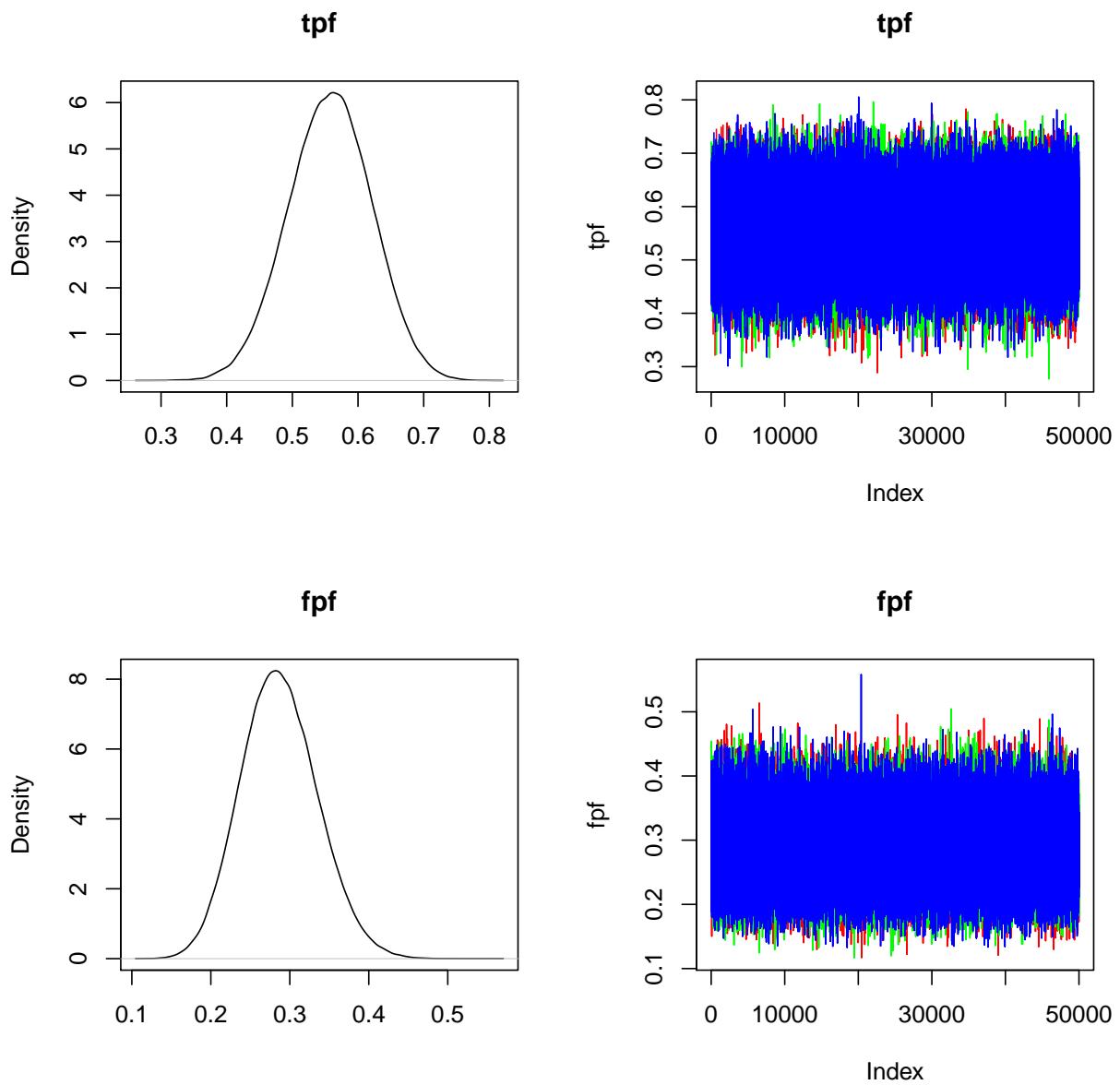
plot(res1$sims.array[,1, i], col=x[1], type = "l", main=J[i],
      ylim=c(min(f)*.99,max(f)*1.01), ylab=J[i])

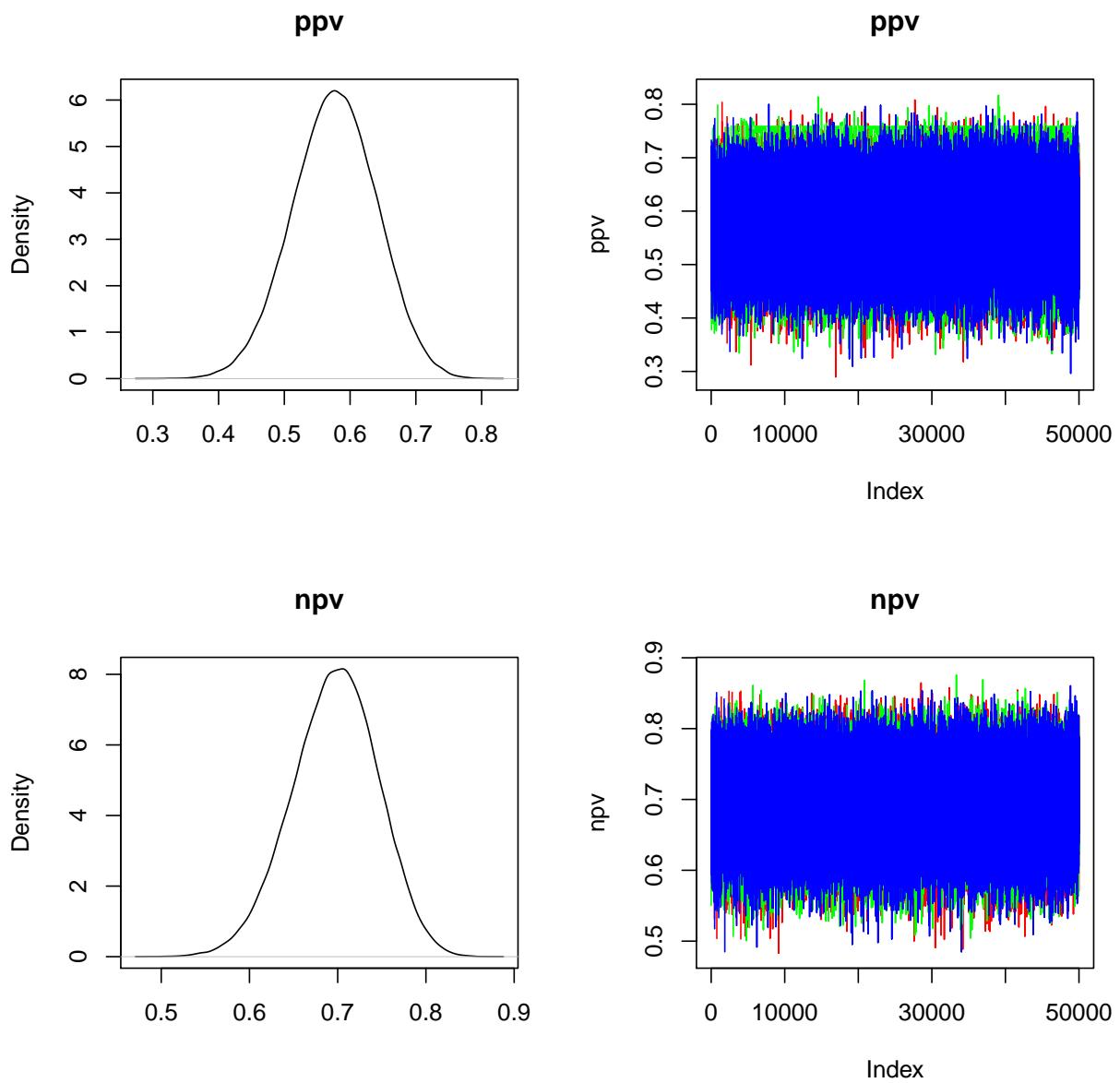
# add the remaining chains
for (ch in 2:(chains)) {
  lines(res1$sims.array[,ch, i], col=x[ch] )
}

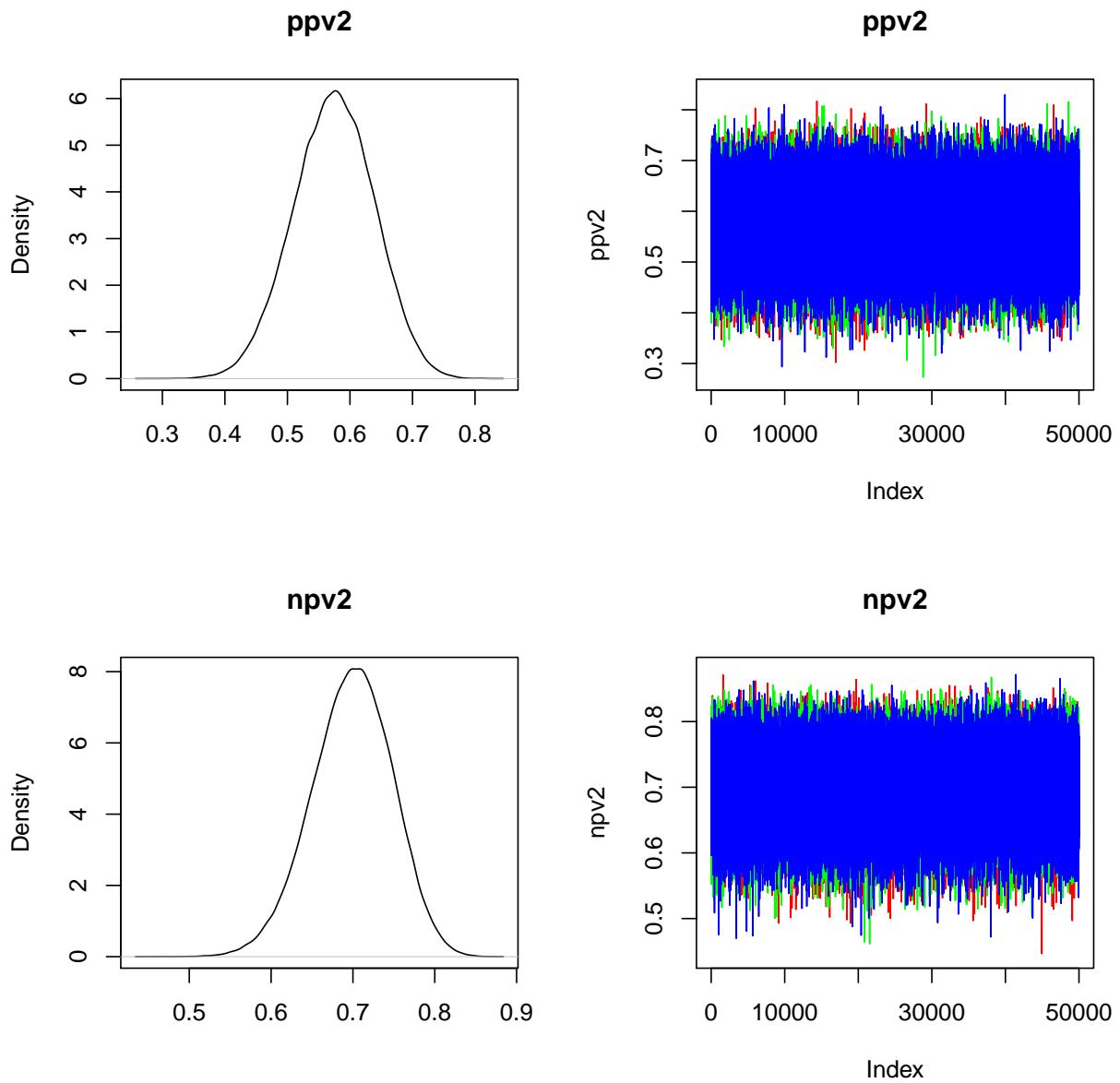
par(mfrow=c(1,1))
}

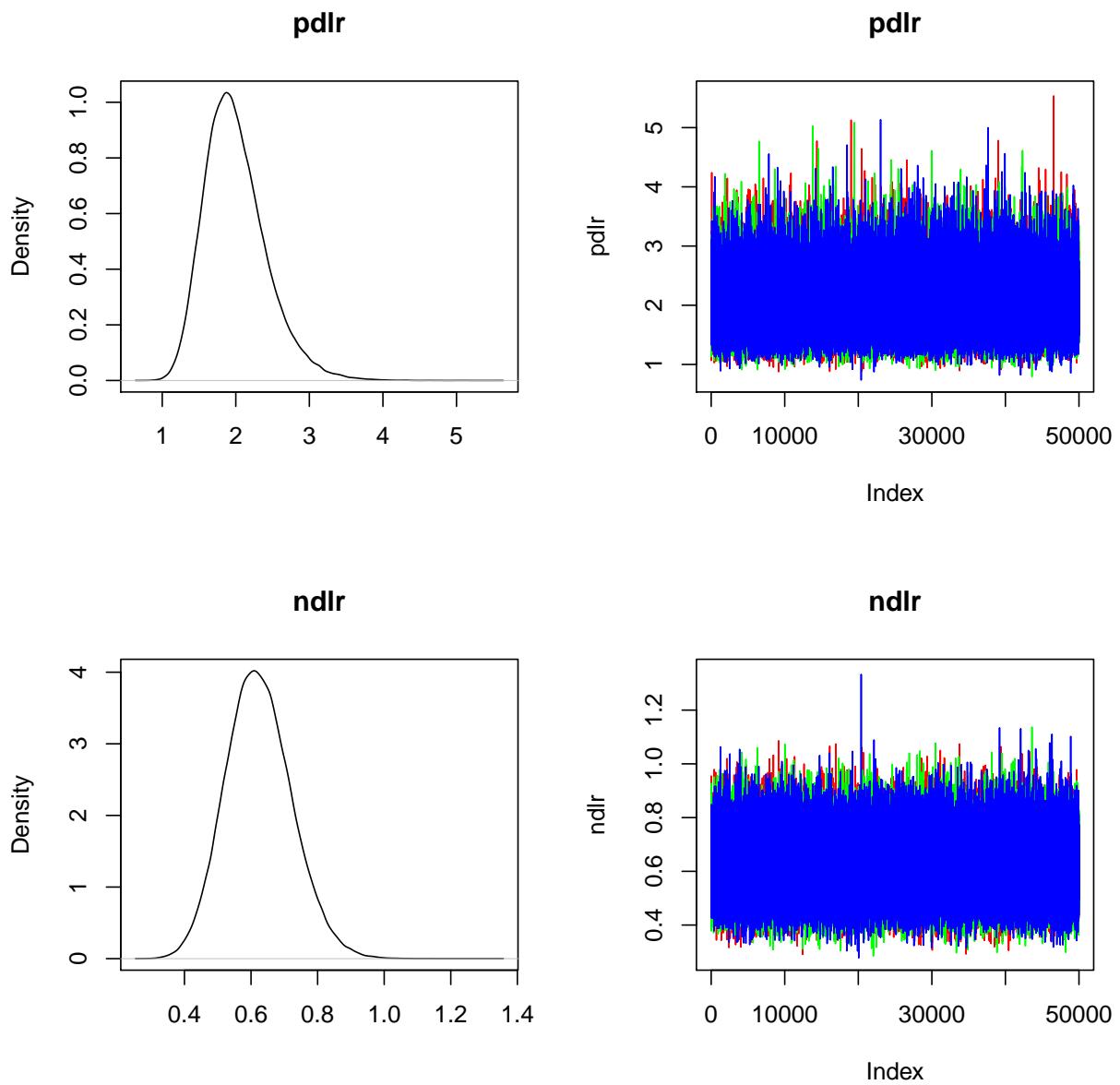
```

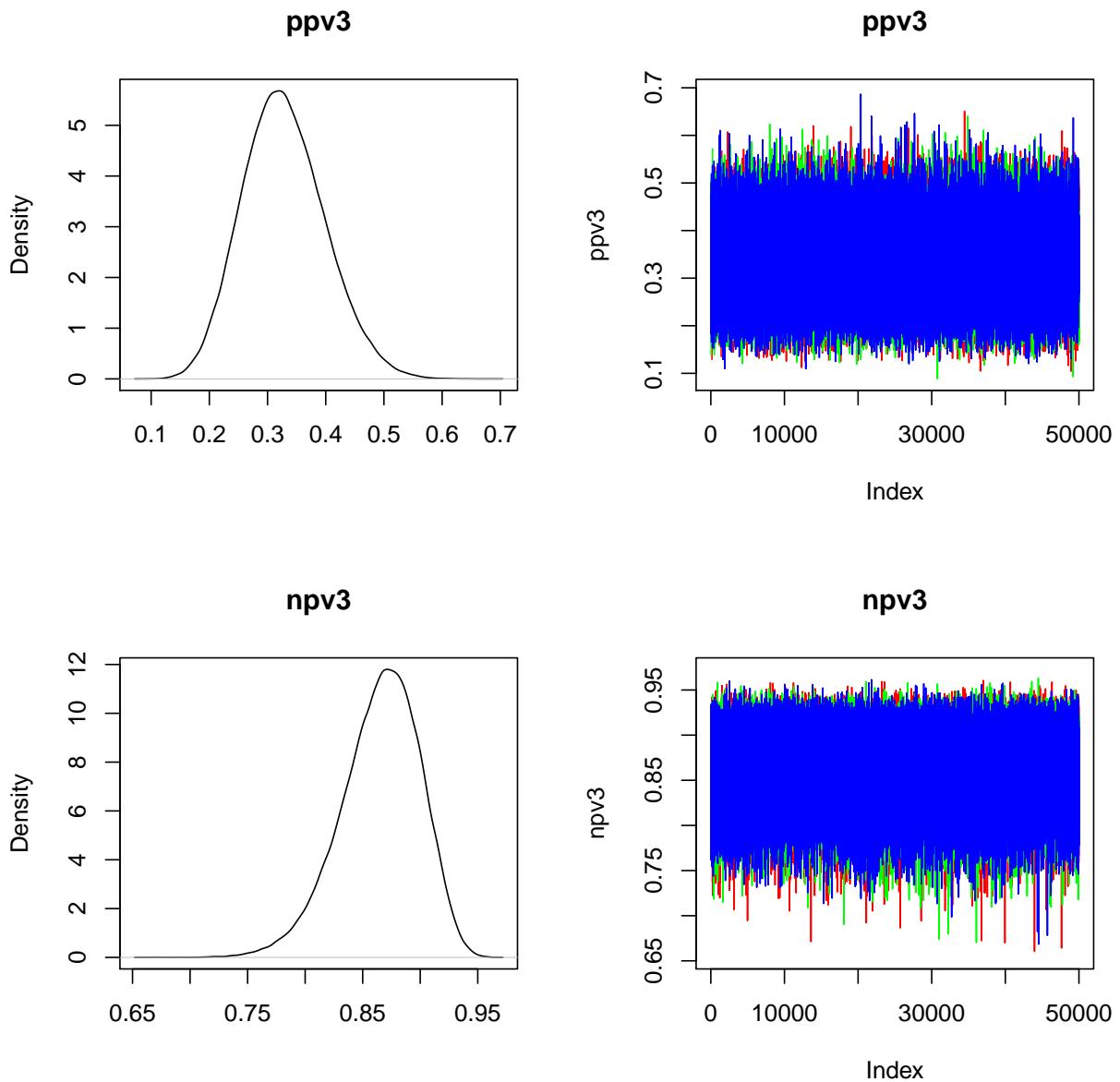












## 17 References

Statistics with Confidence Altman p109 for LR confidence intervals

Bayesian methods in Epidemiology, Lyle D. Broemeling

Advanced Bayesian Methods for Medical Test Accuracy, Lyle D. Broemeling

Applied Bayesian Statistics With R and OpenBUGS Examples

<http://www.australianprescriber.com/magazine/26/5/111/13/>

OpenBUGS ERROR ‘NIL dereference (read)’ solved by turning DIC off: <http://mathstat.helsinki.fi/openbugs/Manuals/TipsTroubleshooting.html>

Warnings:

<http://stats.stackexchange.com/questions/178117/what-happens-with-sensitivity-and-specificity-after-a-second-test/178145#178145>

<http://stats.stackexchange.com/questions/67027/combining-sensitivity-and-specificity-to-measure-classification-performance?rq=1>

"If the method you are using does not yield probabilities I suggest finding another method." Frank Harrell  
Aug 11 '13 at 17:31

## 18 Computing Environment

```
R version 3.2.2 (2015-08-14)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 8 x64 (build 9200)

locale:
[1] LC_COLLATE=English_United Kingdom.1252
[2] LC_CTYPE=English_United Kingdom.1252
[3] LC_MONETARY=English_United Kingdom.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.1252

attached base packages:
[1] stats      graphics   grDevices utils      datasets   methods
[7] base

other attached packages:
[1] LearnBayes_2.15    xtable_1.8-2      R2OpenBUGS_3.2-3.1
[4] epitools_0.5-7     Hmisc_3.17-4      ggplot2_2.1.0
[7] Formula_1.2-1     survival_2.39-5   lattice_0.20-33
[10] binom_1.1-1      knitr_1.13

loaded via a namespace (and not attached):
[1] Rcpp_0.12.6        formatR_1.4       RColorBrewer_1.1-2
[4] plyr_1.8.4         highr_0.6        tools_3.2.2
[7] boot_1.3-17        rpart_4.1-10     digest_0.6.10
[10] evaluate_0.9       gtable_0.2.0     Matrix_1.2-2
[13] yaml_2.1.13        gridExtra_2.2.1   coda_0.18-1
[16] stringr_1.0.0      cluster_2.0.3    grid_3.2.2
[19] nnet_7.3-12        data.table_1.9.6  foreign_0.8-66
[22] rmarkdown_1.0       latticeExtra_0.6-28 magrittr_1.5
[25] scales_0.4.0       htmltools_0.3.5   splines_3.2.2
[28] colorspace_1.2-6   stringi_1.1.1    acepack_1.3-3.3
[31] munsell_0.4.3      chron_2.3-47

[1] "~\\R SCRIPTS\\USEFUL CODE"
```

This took 34.86 seconds to execute.