

Dependent Dirichlet process mixtures

2023-12-14

Single weights dependent Dirichlet process mixture of normal distributions

Linear effect of the covariate

```
require(MASS)
require(splines)
require(pracma)
require(nor1mix)
require(ggplot2)
require(splines)

# Function to compute the ordinary least squares estimates
# (based on the reg model:  $y = x\beta + \epsilon$ ) These are
# to be used as the initial values for the coefficient of
# reg parameters in each component
ols.function <- function(X, y, vcov = FALSE) {
  res <- list()
  if (vcov) {
    res$vcov <- solve(crossprod(X))
    res$coeff <- res$vcov %*% crossprod(X, y)
  } else {
    res$coeff <- try(solve(crossprod(X), crossprod(X, y)),
                     silent = TRUE)
  }
  res
}

bddp <- function(y, X, prior, mcmc, standardise = TRUE) {
  multinom <- function(prob) {
    probs <- t(apply(prob, 1, cumsum))
    res <- rowSums(probs - runif(nrow(probs)) < 0) + 1
    return(res)
  }

  yt <- y
  if (standardise == TRUE) {
    yt <- (y - mean(y))/sd(y)
  }
  n <- length(y)
  k <- ncol(X)
```

```

m <- prior$m0
S <- prior$S0
nu <- prior$nu
psi <- prior$Psi
a <- prior$a
b <- prior$b
alpha <- prior$alpha
L <- prior$L

nburn <- mcmc$nburn
nsave <- mcmc$nsave
nskip <- mcmc$nskip
nsim <- nburn + nsave * nskip

p <- ns <- rep(0, L)
v <- rep(1/L, L)
v[L] <- 1

z <- matrix(NA_real_, nrow = nsim, ncol = n, dimnames = list(1:nsim,
    1:n))
z_tmp <- vector(length = n)

z[1, ] <- rep(1, n)

beta <- matrix(0, nrow = L, ncol = k)
aux <- ols.function(X, yt)$coeff
if (!inherits(aux, "try-error")) {
    for (l in 1:L) {
        beta[l, ] <- aux
    }
}

tau <- rep(1/var(yt), L)
prop <- prob <- matrix(NA_real_, nrow = n, ncol = L)

P <- Tau <- matrix(NA_real_, nrow = nsim, ncol = L, dimnames = list(1:nsim,
    1:L))
Beta <- array(NA_real_, c(nsim, L, k), dimnames = list(1:nsim,
    1:L, colnames(X)))

Beta[1, , ] <- beta
Tau[1, ] <- tau

mu <- mvtnorm(1, mu = m, Sigma = S)
Sigmainv <- rWishart(1, df = nu, solve(nu * psi))[, , 1]

for (i in 2:nsim) {
    cumv <- cumprod(1 - v)
    p[1] <- v[1]
    p[2:L] <- v[2:L] * cumv[1:(L - 1)]

    for (l in 1:L) {
        prop[, l] <- p[l] * dnorm(yt, mean = X %*% beta[l,

```

```

        ] , sd = sqrt(1/tau[1]))
}

prob <- prop/rowSums(prop)

z_tmp <- multinom(prob)

ns <- sapply(1:L, function(x, v) sum(v == x), v = z_tmp)

v[1:(L - 1)] <- rbeta(L - 1, 1 + ns[1:(L - 1)], alpha +
  rev(cumsum(rev(ns[-1]))))

Sigmainv_mu <- Sigmainv %*% mu

for (l in 1:L) {
  X_l <- matrix(X[z_tmp == l, ], ncol = k, nrow = ns[l])
  yt_l <- yt[z_tmp == l]
  V <- solve(Sigmainv + tau[l] * crossprod(X_l))
  mu1 <- V %*% (Sigmainv_mu + tau[l] * crossprod(X_l,
    yt_l))
  beta[l, ] <- mvrnorm(1, mu = mu1, Sigma = V)

  aux <- yt_l - X_l %*% beta[l, ]
  tau[l] <- rgamma(1, shape = a + (ns[l]/2), rate = b +
    0.5 * crossprod(aux))
}

S_inv <- solve(S)
Vaux <- solve(S_inv + L * Sigmainv)
if (k == 1) {
  meanmu <- Vaux %*% (S_inv %*% m + Sigmainv %*% sum(beta))
} else {
  meanmu <- Vaux %*% (S_inv %*% m + Sigmainv %*% colSums(beta))
}
mu <- mvrnorm(1, mu = meanmu, Sigma = Vaux)

Vaux1 <- 0
for (l in 1:L) {
  Vaux1 <- Vaux1 + tcrossprod(beta[l, ] - mu)
}

Sigmainv <- rWishart(1, nu + L, solve(nu * psi + Vaux1))[, ,
  1]

P[i, ] <- p
z[i, ] <- z_tmp
Beta[i, , ] <- beta
Tau[i, ] <- tau
}

if (standardise == TRUE) {
  Beta[, , 1] <- sd(y) * Beta[, , 1] + mean(y)
  if (k > 1) {

```

```

        Beta[, , 2:k] <- sd(y) * Beta[, , 2:k]
    }
    Sigma2 <- var(y) * (1/Tau)
} else {
    Sigma2 <- 1/Tau
}

res <- list()
res$z <- z[seq(nburn + 1, nsim, by = nskip), ]
res$P <- P[seq(nburn + 1, nsim, by = nskip), ]
res$Beta <- Beta[seq(nburn + 1, nsim, by = nskip), , , drop = FALSE]
res$Sigma2 <- Sigma2[seq(nburn + 1, nsim, by = nskip), ]
res
}

# Generate simulated data
set.seed(123)
n <- 1000
x <- runif(n, -5, 5)
y <- rnorm(n, x^2, 1)
X <- cbind(rep(1, n), x)

prior <- list(m0 = rep(0, 2), S0 = 10 * diag(2), nu = 4, Psi = diag(2),
a = 2, b = 2, alpha = 1, L = 10)

mcmc <- list(nburn = 1000, nsave = 5000, nskip = 1)

# the option standardise equal to TRUE allows to
# standardise the test outcomes
set.seed(123)
fit <- bddp(y = y, X = X, prior = prior, mcmc = mcmc, standardise = TRUE)

# grid along the covariate range to predict the mean
# function
xpred1 <- seq(min(x), max(x), length = 100)
npred1 <- length(xpred1)
Xp1 <- cbind(rep(1, npred1), xpred1)

meanfun <- matrix(0, nrow = npred1, ncol = mcmc$nsave)
for (l in 1:mcmc$nsave) {
    for (j in 1:npred1) {
        meanfun[j, l] <- sum(fit$P[, ] * Xp1[j, ] %*% t(fit$Beta[, ,
            ]))
    }
}

meanfunm <- apply(meanfun, 1, median)
meanfunl <- apply(meanfun, 1, quantile, prob = 0.025)
meanfuh <- apply(meanfun, 1, quantile, prob = 0.975)

dfdata <- data.frame(x = x, y = y)
meant <- xpred1^2

```

```

mean.plot <- data.frame(regm = meanfunm, regl = meanfunl, regh = meanfunh,
                        xpred = xpred1, regt = meant)

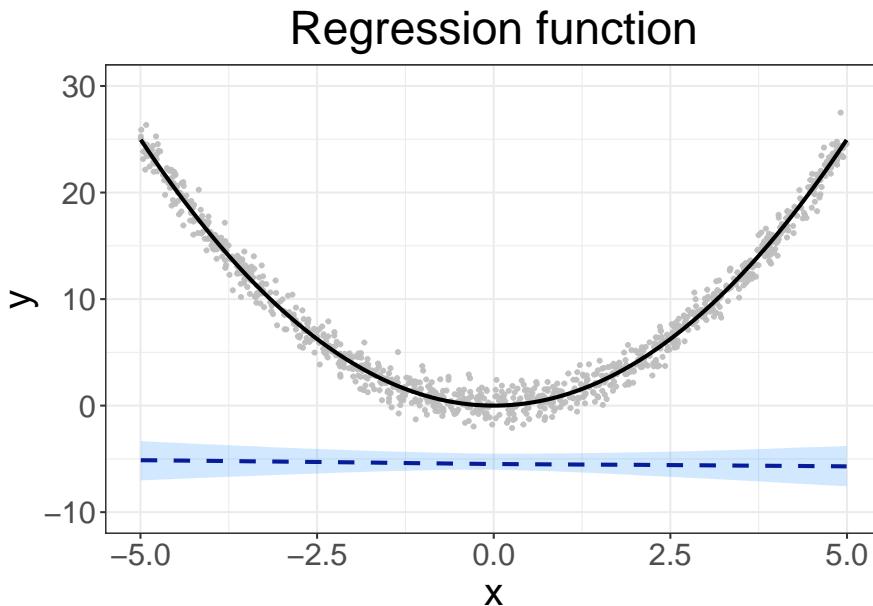
ggplot(dfdata, aes(x = x, y = y)) + geom_point(alpha = 0.95,
                                                cex = 0.8, col = "grey") + geom_line(data = mean.plot, aes(x = xpred,
                                                y = regm), size = 1, col = "navyblue", linetype = "dashed") +
  geom_ribbon(data = mean.plot, aes(x = xpred, ymin = regl,
                                    ymax = regh), alpha = 0.2, inherit.aes = FALSE, fill = "dodgerblue1") +
  geom_line(data = mean.plot, aes(x = xpred, y = regt), size = 1) +
  theme_bw() + theme(text = element_text(size = 20)) + ggtitle("Regression function") +
  theme(plot.title = element_text(hjust = 0.5)) + ylim(-10,
  30)

```

```

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```



```

# grids along y and x to predict the conditional density
xpred <- seq(-5, 5, by = 0.5)
npred <- length(xpred)
Xp <- cbind(rep(1, npred), xpred)
grid <- seq(min(y) - 1, max(y) + 1, len = 100)
ngrid <- length(grid)

dpred <- array(0, c(mcmc$nsave, ngrid, npred))

for (k in 1:npred) {
  for (l in 1:mcmc$nsave) {
    for (j in 1:ngrid) {
      dpred[l, j, k] <- sum(fit$P[l, ] * dnorm(grid[j],
                                                mean = Xp[k, ] %*% t(fit$Beta[1, , ])), sd = sqrt(fit$Sigma2[1,

```

```

        ])))
    }
}

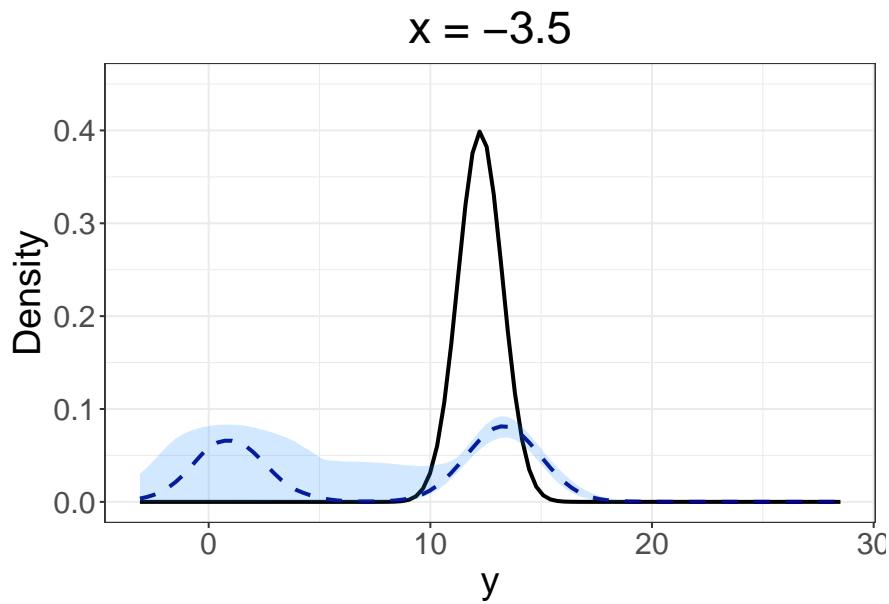
p.dpred <- ql.dpred <- qh.dpred <- matrix(0, nrow = ngrid, ncol = npred)
for (k in 1:npred) {
  for (j in 1:ngrid) {
    p.dpred[j, k] <- median(dpred[, j, k])
    ql.dpred[j, k] <- quantile(dpred[, j, k], 0.025)
    qh.dpred[j, k] <- quantile(dpred[, j, k], 0.975)
  }
}

# true conditional density
dtrue <- function(grid, x) {
  dnorm(grid, mean = x^2, sd = 1)
}

# plot two profiles of the conditional density
dt1 <- dtrue(grid, xpred[4])
dfpred1 <- data.frame(dm = p.dpred[, 4], dl = ql.dpred[, 4],
  dh = qh.dpred[, 4], seqgrid = grid, dt = dt1)

ggplot() + geom_line(data = dfpred1, aes(x = seqgrid, y = dt),
  colour = "black", size = 1) + geom_line(data = dfpred1, aes(x = seqgrid,
  y = dm), col = "navyblue", linetype = "dashed", size = 1) +
  geom_ribbon(data = dfpred1, aes(x = seqgrid, ymin = dl, ymax = dh),
  alpha = 0.2, fill = "dodgerblue1") + theme_bw() + xlab("y") +
  ylab("Density") + theme(text = element_text(size = 20)) +
  ggtitle("x = -3.5") + theme(plot.title = element_text(hjust = 0.5)) +
  ylim(0, 0.45)

```

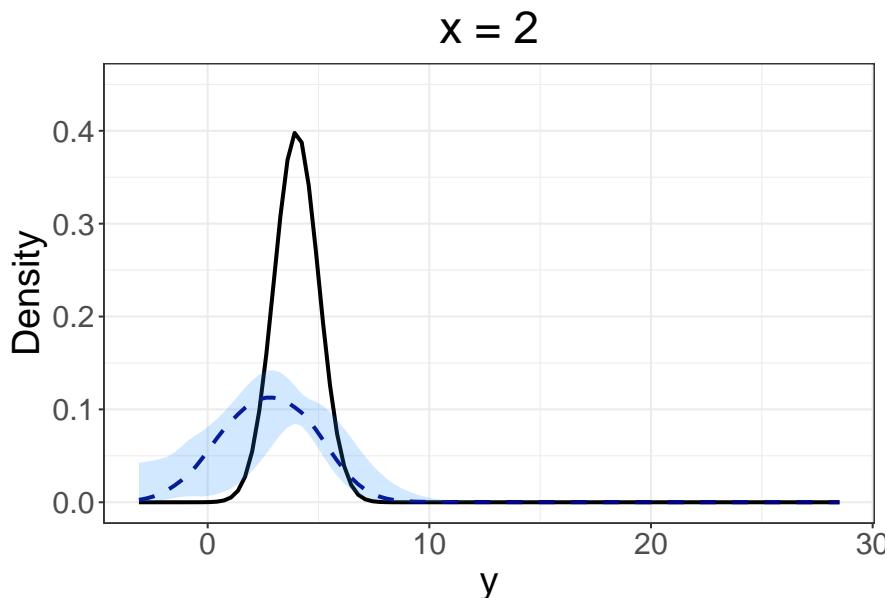


```

dt2 <- dtrue(grid, xpred[15])
dfpred2 <- data.frame(dm = p.dpred[, 15], dl = ql.dpred[, 15],
                      dh = qh.dpred[, 15], seqgrid = grid, dt = dt2)

ggplot() + geom_line(data = dfpred2, aes(x = seqgrid, y = dt),
                      colour = "black", size = 1) + geom_line(data = dfpred2, aes(x = seqgrid,
                      y = dm), col = "navyblue", linetype = "dashed", size = 1) +
  geom_ribbon(data = dfpred2, aes(x = seqgrid, ymin = dl, ymax = dh),
               alpha = 0.2, fill = "dodgerblue1") + theme_bw() + xlab("y") +
  ylab("Density") + theme(text = element_text(size = 20)) +
  ggtitle("x = 2") + theme(plot.title = element_text(hjust = 0.5)) +
  ylim(0, 0.45)

```



Effect of the covariate modelled through cubic B-spline basis functions

```

# Cubic B-spline basis with no interior knots
X <- cbind(rep(1, n), bs(x, degree = 3, knots = c(), intercept = FALSE))

prior <- list(m0 = rep(0, 4), S0 = 10 * diag(4), nu = 6, Psi = diag(4),
              a = 2, b = 2, alpha = 1, L = 10)

mcmc <- list(nburn = 1000, nsave = 5000, nskip = 1)

set.seed(123)
fit <- bddp(y = y, X = X, prior = prior, mcmc = mcmc, standardise = TRUE)

xpred1 <- seq(min(x), max(x), length = 100)
npred1 <- length(xpred1)
Xp1 <- cbind(rep(1, npred1), predict(bs(x, degree = 3, knots = c(),
                                         intercept = FALSE), xpred1))

```

```

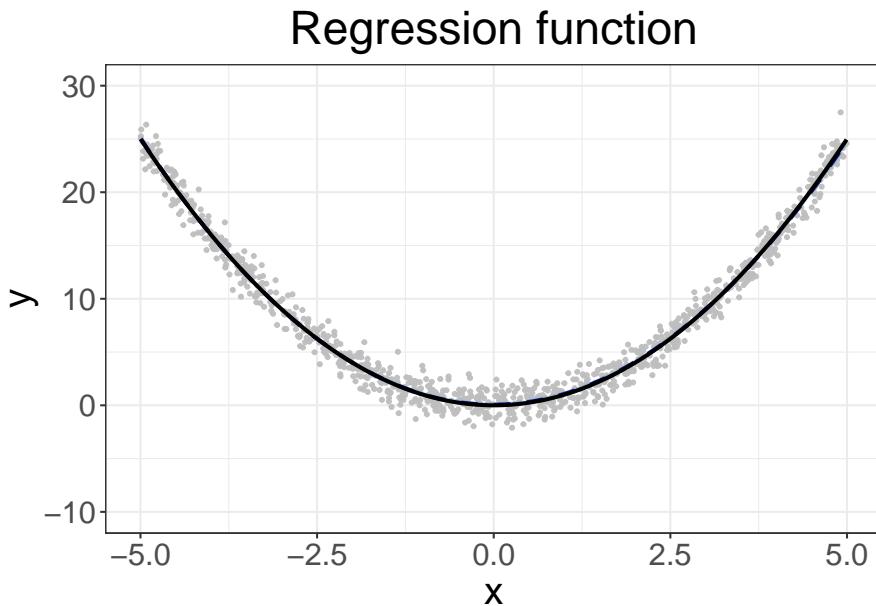
meanfun <- matrix(0, nrow = npred1, ncol = mcmc$nsave)
for (l in 1:mcmc$nsave) {
  for (j in 1:npred1) {
    meanfun[j, l] <- sum(fit$P[1, ] * Xp1[j, ] %*% t(fit$Beta[l,
      , ]))
  }
}

meanfunm <- apply(meanfun, 1, median)
meanfunl <- apply(meanfun, 1, quantile, prob = 0.025)
meanfuh <- apply(meanfun, 1, quantile, prob = 0.975)

dfdata <- data.frame(x = x, y = y)
meant <- xpred1^2
mean.plot <- data.frame(regm = meanfunm, regl = meanfunl, regh = meanfuh,
  xpred = xpred1, regt = meant)

ggplot(dfdata, aes(x = x, y = y)) + geom_point(alpha = 0.95,
  cex = 0.8, col = "grey") + geom_line(data = mean.plot, aes(x = xpred,
  y = regm), size = 1, col = "navyblue", linetype = "dashed") +
  geom_ribbon(data = mean.plot, aes(x = xpred, ymin = regl,
  ymax = regh), alpha = 0.2, inherit.aes = FALSE, fill = "dodgerblue1") +
  geom_line(data = mean.plot, aes(x = xpred, y = regt), size = 1) +
  theme_bw() + theme(text = element_text(size = 20)) + ggtitle("Regression function") +
  theme(plot.title = element_text(hjust = 0.5)) + ylim(-10,
  30)

```



```

xpred <- seq(-5, 5, by = 0.5)
npred <- length(xpred)
Xp <- cbind(rep(1, npred), predict(bs(x, degree = 3, knots = c(),
  intercept = FALSE), xpred))

```

```
## Warning in bs(x = c(-5, -4.5, -4, -3.5, -3, -2.5, -2, -1.5, -1, -0.5, 0, : some
```

```

## 'x' values beyond boundary knots may cause ill-conditioned bases

grid <- seq(min(y) - 1, max(y) + 1, len = 100)
ngrid <- length(grid)

dpred <- array(0, c(mcmc$nsave, ngrid, npred))

for (k in 1:npred) {
  for (l in 1:mcmc$nsave) {
    for (j in 1:ngrid) {
      dpred[l, j, k] <- sum(fit$P[, ] * dnorm(grid[j],
        mean = Xp[, ] %*% t(fit$Beta[, , ])), sd = sqrt(fit$Sigma2[, ,
        ])))
    }
  }
}

p.dpred <- ql.dpred <- qh.dpred <- matrix(0, nrow = ngrid, ncol = npred)
for (k in 1:npred) {
  for (j in 1:ngrid) {
    p.dpred[j, k] <- median(dpred[, j, k])
    ql.dpred[j, k] <- quantile(dpred[, j, k], 0.025)
    qh.dpred[j, k] <- quantile(dpred[, j, k], 0.975)
  }
}

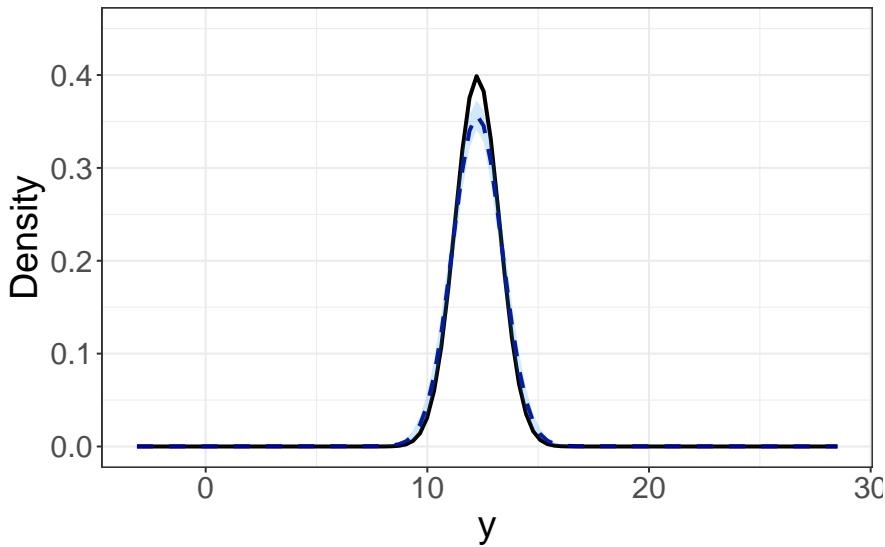
dtrue <- function(grid, x) {
  dnorm(grid, mean = x^2, sd = 1)
}

dt1 <- dtrue(grid, xpred[4])
dfpred1 <- data.frame(dm = p.dpred[, 4], dl = ql.dpred[, 4],
  dh = qh.dpred[, 4], seqgrid = grid, dt = dt1)

ggplot() + geom_line(data = dfpred1, aes(x = seqgrid, y = dt),
  colour = "black", size = 1) + geom_line(data = dfpred1, aes(x = seqgrid,
  y = dm), col = "navyblue", linetype = "dashed", size = 1) +
  geom_ribbon(data = dfpred1, aes(x = seqgrid, ymin = dl, ymax = dh),
  alpha = 0.2, fill = "dodgerblue1") + theme_bw() + xlab("y") +
  ylab("Density") + theme(text = element_text(size = 20)) +
  ggtitle("x = -3.5") + theme(plot.title = element_text(hjust = 0.5)) +
  ylim(0, 0.45)

```

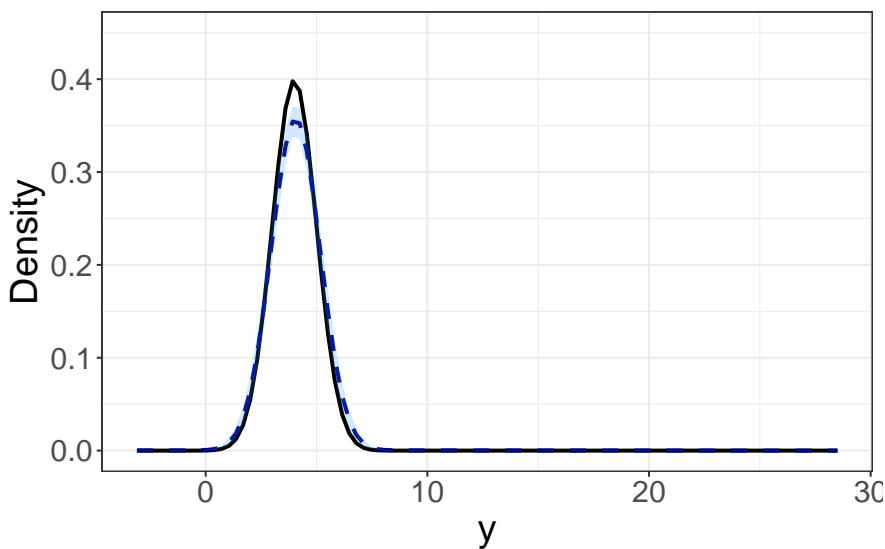
$x = -3.5$



```
dt2 <- dtrue(grid, xpred[15])
dfpred2 <- data.frame(dm = p.dpred[, 15], dl = ql.dpred[, 15],
                      dh = qh.dpred[, 15], seqgrid = grid, dt = dt2)

ggplot() + geom_line(data = dfpred2, aes(x = seqgrid, y = dt),
                      colour = "black", size = 1) + geom_line(data = dfpred2, aes(x = seqgrid,
                      y = dm), col = "navyblue", linetype = "dashed", size = 1) +
  geom_ribbon(data = dfpred2, aes(x = seqgrid, ymin = dl, ymax = dh),
              alpha = 0.2, fill = "dodgerblue1") + theme_bw() + xlab("y") +
  ylab("Density") + theme(text = element_text(size = 20)) +
  ggtitle("x = 2") + theme(plot.title = element_text(hjust = 0.5)) +
  ylim(0, 0.45)
```

$x = 2$



Application to ROC regression

```

library(ROCnReg)
library(gridExtra)
library(grid)
library(ggplot2)
library(coda)
library(normmix)
require(coda)

data("endosyn")

# Data frame for predicting the covariate-specific ROC
# curve
agep <- seq(22, 80, l = 30)
endopred <- data.frame(age = rep(agep, 2), gender = factor(rep(c("Women",
  "Men"), each = length(agep)))))

# Bayesian approach with L = 1 and parametric/linear
# effects set.seed(123, 'L'Ecuyer-CMRG') # for
# reproducibility
cROC_bp <- cROC.bnp(formula.h = bmi ~ gender * age, formula.d = bmi ~
  gender * age, group = "cvd_idf", tag.h = 0, data = endosyn,
  newdata = endopred, standardise = TRUE, p = seq(0, 1, l = 101),
  ci.level = 0.95, compute.lpml = TRUE, compute.WAIC = TRUE,
  compute.DIC = TRUE, pauc = paucontrol(compute = FALSE),
  prior.h = priorcontrol.bnp(L = 1), prior.d = priorcontrol.bnp(L = 1),
  density = densitycontrol(compute = TRUE), mcmc = mcmccontrol(nsave = 8000,
  nburn = 2000, nskip = 1), parallel = "multicore", ncpus = 10)

summary(cROC_bp)

##
## Call:
## cROC.bnp(formula.h = bmi ~ gender * age, formula.d = bmi ~ gender *
##   age, group = "cvd_idf", tag.h = 0, data = endosyn, newdata = endopred,
##   standardise = TRUE, p = seq(0, 1, l = 101), ci.level = 0.95,
##   compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE,
##   pauc = paucontrol(compute = FALSE), density = densitycontrol(compute = TRUE),
##   prior.h = priorcontrol.bnp(L = 1), prior.d = priorcontrol.bnp(L = 1),
##   mcmc = mcmccontrol(nsave = 8000, nburn = 2000, nskip = 1),
##   parallel = "multicore", ncpus = 10)
##
## Approach: Conditional ROC curve - Bayesian nonparametric
## -----
## 
## Parametric coefficients
## Group H:
##             Post. mean    Post. quantile 2.5%    Post. quantile 97.5%
## (Intercept) 26.1480          25.8735          26.4133
## genderWomen -0.9168          -1.2658          -0.5638
## age          1.1982           0.9251           1.4743

```

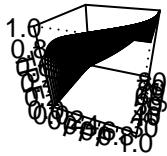
```

## genderWomen:age      1.1908          0.8448          1.5405
##
##
## Group D:
##           Post. mean   Post. quantile 2.5%   Post. quantile 97.5%
## (Intercept)    29.1894        28.7625        29.6219
## genderWomen     2.0777        1.3911        2.7643
## age            0.6575        0.2283        1.0927
## genderWomen:age -0.7661       -1.4273       -0.1037
##
##
## ROC curve:
##           Post. mean   Post. quantile 2.5%   Post. quantile 97.5%
## (Intercept)   -0.6962       -0.8167       -0.5763
## genderWomen   -0.6855       -0.8646       -0.5067
## age           0.1238        0.0059        0.2411
## genderWomen:age 0.4480        0.2697        0.6233
## b              0.9396        0.8829        0.9968
##
##
## Model selection criteria:
##           Group H     Group D
## WAIC      12175.021    4007.822
## WAIC (Penalty) 6.304      5.569
## LPML     -6087.511   -2003.912
## DIC      12173.698    4007.159
## DIC (Penalty) 5.009      4.969
##
##
## Sample sizes:
##           Group H     Group D
## Number of observations 2149      691
## Number of missing data  0         0

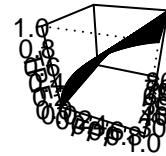
op <- par(mfrow = c(2, 2))
plot(cROC_bp, ask = FALSE)

```

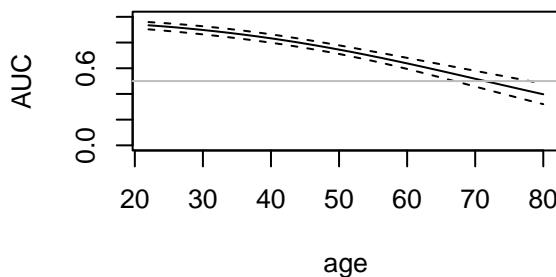
Conditional ROC surface
Women



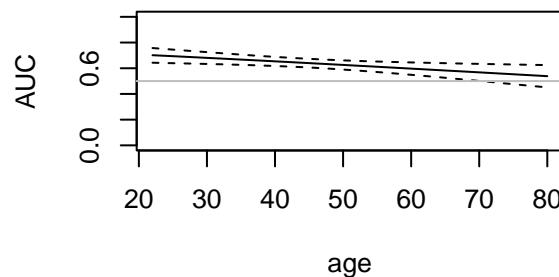
Conditional ROC surface
Men



Area under the conditional ROC curve
gender = Women



Area under the conditional ROC curve
gender = Men



```
par(op)
```

```
# Bayesian approach with L = 10 and nonlinear effects of
# age Note: Time consuming
```

```
set.seed(123, "L'Ecuyer-CMRG") # for reproducibility
levels(endosyn$gender)
```

```
## [1] "Men"    "Women"
```

```
cROC_bnp <- cROC.bnp(formula.h = bmi ~ gender + f(age, by = gender,
K = c(0, 0)), formula.d = bmi ~ gender + f(age, by = gender,
K = c(4, 4)), group = "cvd_idf", tag.h = 0, data = endosyn,
newdata = endopred, standardise = TRUE, p = seq(0, 1, l = 101),
ci.level = 0.95, compute.lpml = TRUE, compute.WAIC = TRUE,
compute.DIC = TRUE, pauc = pauccontrol(compute = FALSE),
prior.h = priorcontrol.bnp(L = 10), prior.d = priorcontrol.bnp(L = 10),
density = densitycontrol(compute = TRUE), mcmc = mcmccontrol(nsave = 8000,
nburn = 2000, nskip = 1), parallel = "multicore", ncpus = 10)
summary(cROC_bnp)
```

```
##
## Call:
## cROC.bnp(formula.h = bmi ~ gender + f(age, by = gender, K = c(0,
## 0)), formula.d = bmi ~ gender + f(age, by = gender, K = c(4,
## 4)), group = "cvd_idf", tag.h = 0, data = endosyn, newdata = endopred,
```

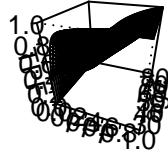
```

## standardise = TRUE, p = seq(0, 1, l = 101), ci.level = 0.95,
## compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE,
## pauc = paucontrol(compute = FALSE), density = densitycontrol(compute = TRUE),
## prior.h = priorcontrol.bnp(L = 10), prior.d = priorcontrol.bnp(L = 10),
## mcmc = mcmccontrol(nsave = 8000, nburn = 2000, nskip = 1),
## parallel = "multicore", ncpus = 10)
##
## Approach: Conditional ROC curve - Bayesian nonparametric
## -----
## 
## Model selection criteria:
##                               Group H      Group D
## WAIC                  11836.341    3910.856
## WAIC (Penalty)        33.451      41.631
## LPML                 -5918.427   -1955.921
## DIC                   11833.018    3904.750
## DIC (Penalty)         31.790      38.578
##
## 
## Sample sizes:
##                               Group H      Group D
## Number of observations     2149       691
## Number of missing data      0          0

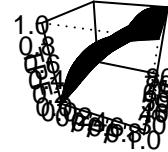
op <- par(mfrow = c(2, 2))
plot(cROC_bnp, ask = FALSE)

```

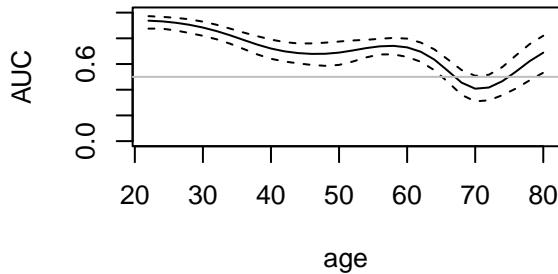
**Conditional ROC surface
Women**



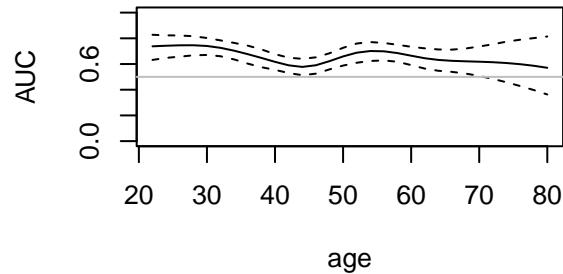
**Conditional ROC surface
Men**



**Area under the conditional ROC curve
gender = Women**



**Area under the conditional ROC curve
gender = Men**

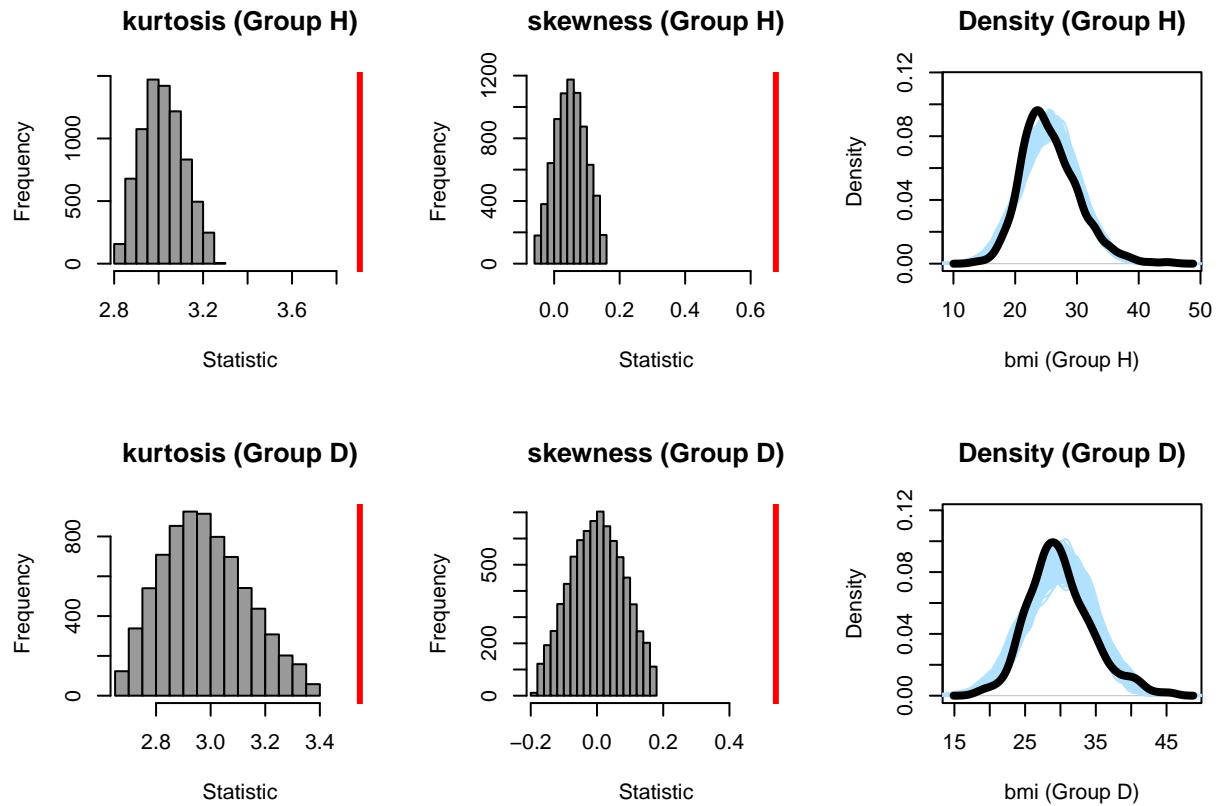


```

par(op)

# Predictive checks
op <- par(mfrow = c(2, 3))
pc_cROC_bp <- predictive.checks(cROC_bp, statistics = c("kurtosis",
  "skewness"), devnew = FALSE)

```

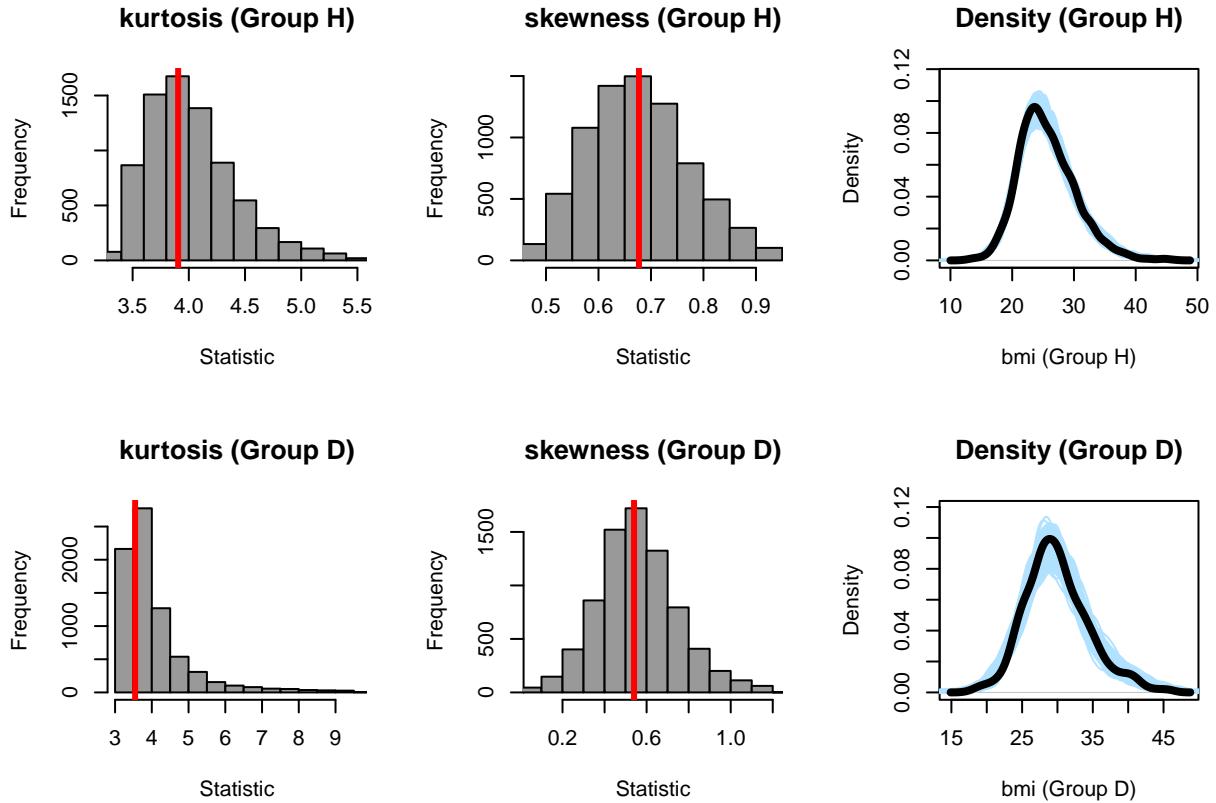


```

par(op)

op <- par(mfrow = c(2, 3))
pc_cROC_bnp <- predictive.checks(cROC_bnp, statistics = c("kurtosis",
  "skewness"), devnew = FALSE)

```



```

par(op)

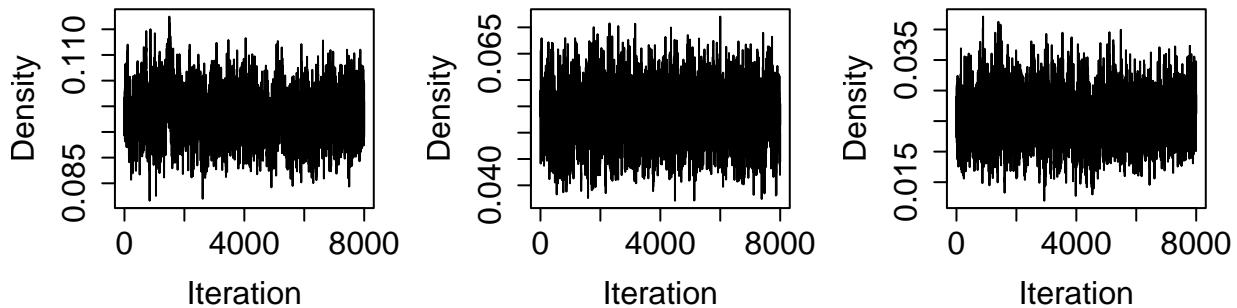
# Trace plots, effective sample sizes and Geweke statistics
# cROC_bnp object

# Trace plots for several BMI values and covariates
# (randomly selected), for healthy and diseased population
# Nondiseased
set.seed(54321, "Mersenne-Twister") # for reproducibility
# BMI values
pos_h_den <- sort(sample(1:length(cROC_bnp$dens$h$grid), 2))
bmi_h_den <- round(cROC_bnp$dens$h$grid[pos_h_den])
# Covariate values
cov_h_den <- sort(sample(1:nrow(cROC_bnp$newdata), 3))

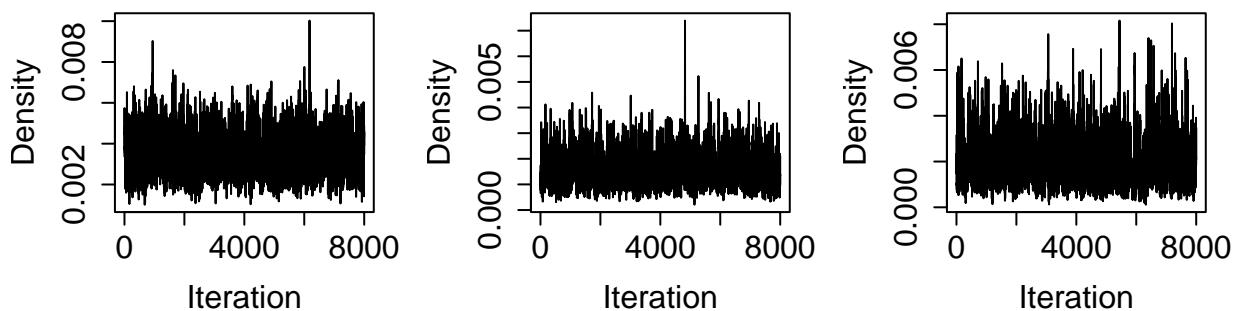
op <- par(mfrow = c(2, 3))
for (i in 1:2) {
  for (j in 1:3) {
    plot(1:cROC_bnp$mcmc$nsave, cROC_bnp$dens$h$dens[pos_h_den[i],
      , cov_h_den[j]], type = "l", xlab = "Iteration",
      ylab = "Density", main = paste0("Trace plot - Nondiseased population \n BMI = ",
      round(bmi_h_den[i]), " - Age = ", cROC_bnp$newdata$age[cov_h_den[j]],
      " - Gender = ", cROC_bnp$newdata$gender[cov_h_den[j]]),
      cex.main = 1.5, cex.lab = 1.5, cex.axis = 1.5)
  }
}

```

ce plot – Nondiseased popuce plot – Nondiseased popuce plot – Nondiseased popu
= 22 – Age = 40 – Gender =
MI = 22 – Age = 38 – Gender
MI = 22 – Age = 52 – Gender



ce plot – Nondiseased popuce plot – Nondiseased popuce plot – Nondiseased popu
= 39 – Age = 40 – Gender =
MI = 39 – Age = 38 – Gender
MI = 39 – Age = 52 – Gender



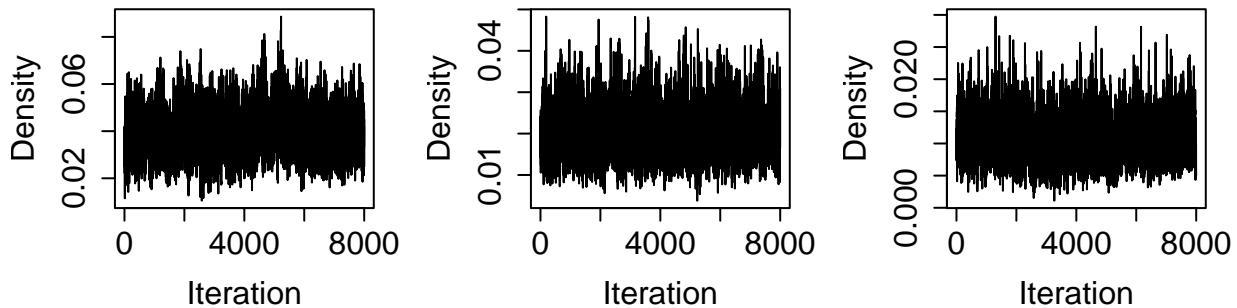
```
par(op)

# Diseased
set.seed(123456, "Mersenne-Twister")
# BMI values
pos_d_den <- sort(sample(1:length(cROC_bnp$dens$d$grid), 2))
bmi_d_den <- round(cROC_bnp$dens$d$grid[pos_d_den])

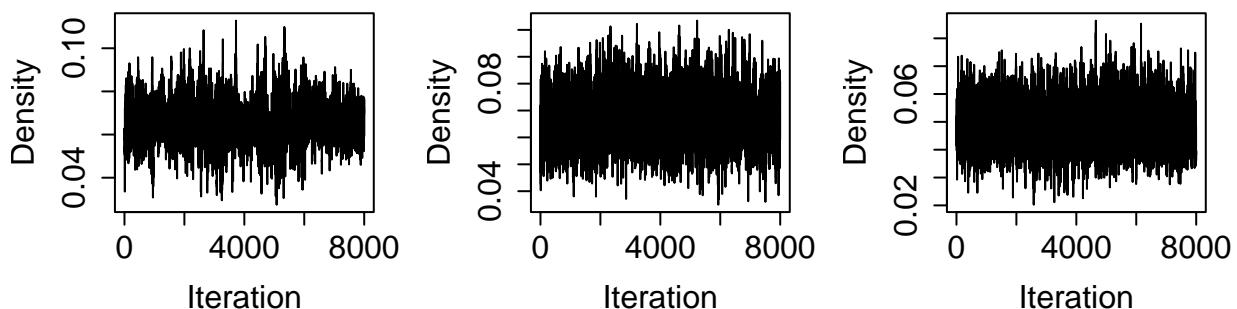
# Covariate values
cov_d_den <- sort(sample(1:nrow(cROC_bnp$newdata), 3))

op <- par(mfrow = c(2, 3))
for (i in 1:2) {
  for (j in 1:3) {
    plot(1:cROC_bnp$mcmc$nsave, cROC_bnp$dens$d$dens[pos_d_den[i],
      , cov_d_den[j]], type = "l", xlab = "Iteration",
      ylab = "Density", main = paste0("Trace plot - Diseased population \n BMI = ",
      round(bmi_d_den[i]), " - Age = ", cROC_bnp$newdata$age[cov_d_den[j]],
      " - Gender = ", cROC_bnp$newdata$gender[cov_d_den[j]]),
      cex.main = 1.5, cex.lab = 1.5, cex.axis = 1.5)
  }
}
```

race plot – Diseased popula
= 23 – Age = 34 – Gender =
MI = 23 – Age = 50 – Gender
MI = 23 – Age = 58 – Gender

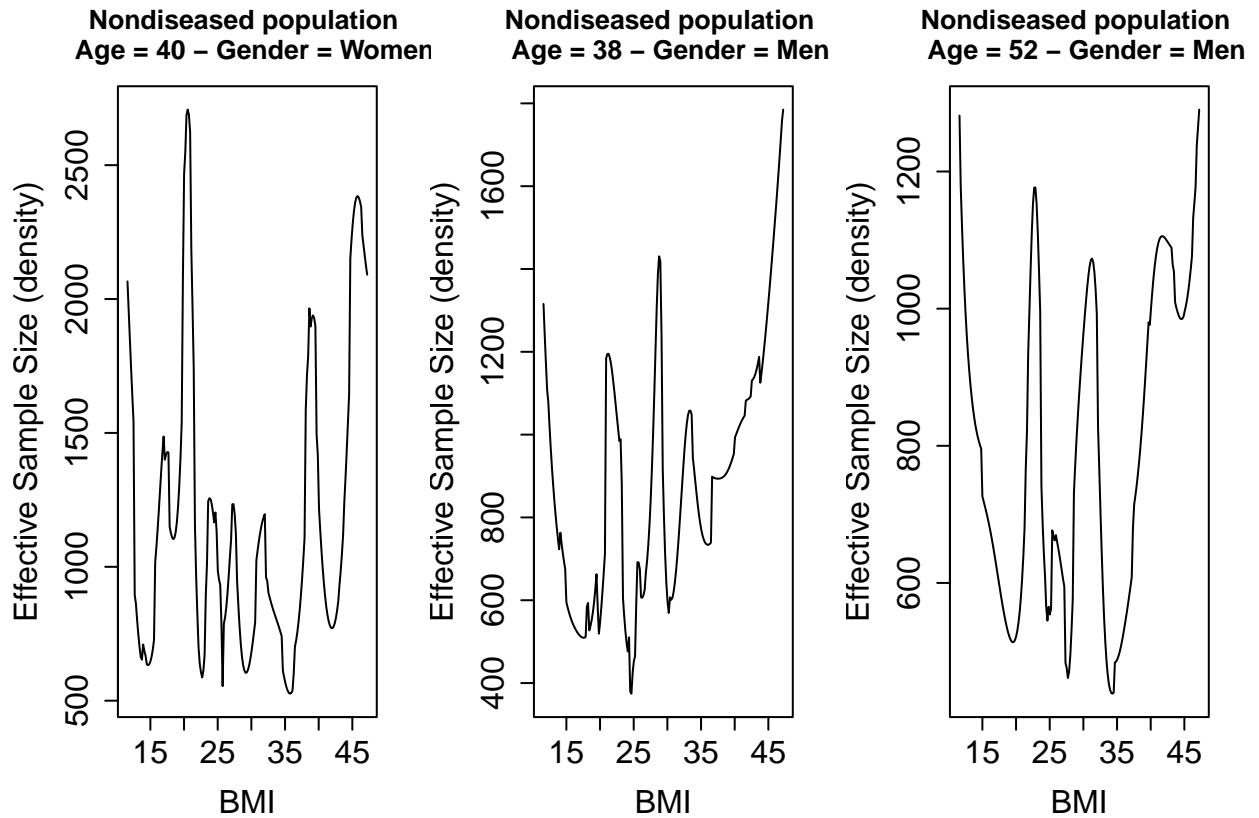


race plot – Diseased popula
= 26 – Age = 34 – Gender =
MI = 26 – Age = 50 – Gender
MI = 26 – Age = 58 – Gender



```
par(op)

# Effective sample size for several covariate values
# (randomly selected) in healthy and diseased populations
# Nondiseased
op <- par(mfrow = c(1, 3))
for (i in 1:3) {
  plot(cROC_bnp$dens$h$grid, effectiveSize(t(cROC_bnp$dens$h$dens[, ,
    , cov_h_den[i]])), type = "l", xlab = "BMI", ylab = "Effective Sample Size (density)",
    main = paste0("Nondiseased population \n Age = ", cROC_bnp$newdata$age[cov_h_den[i]],
      " - Gender = ", cROC_bnp$newdata$gender[cov_h_den[i]]),
    cex.main = 1.2, cex.lab = 1.5, cex.axis = 1.5)
}
```

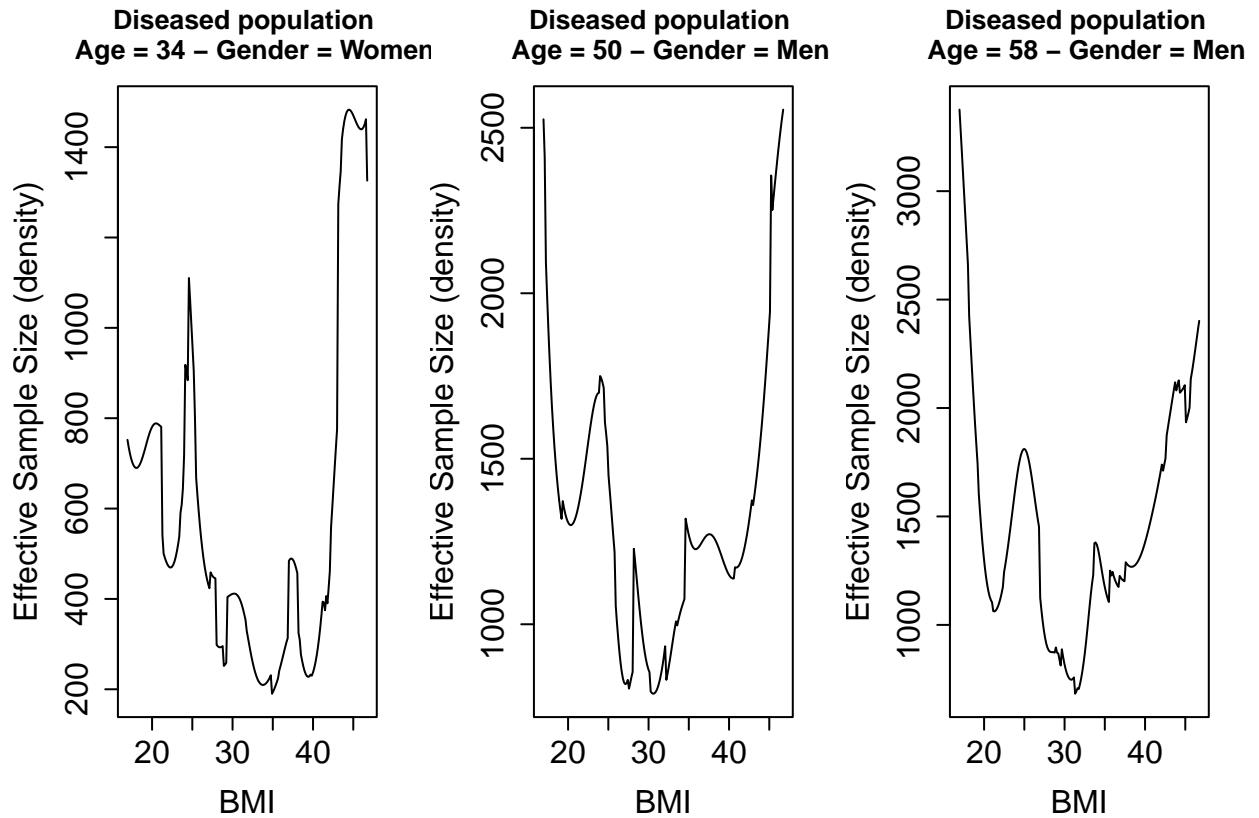


```

par(op)

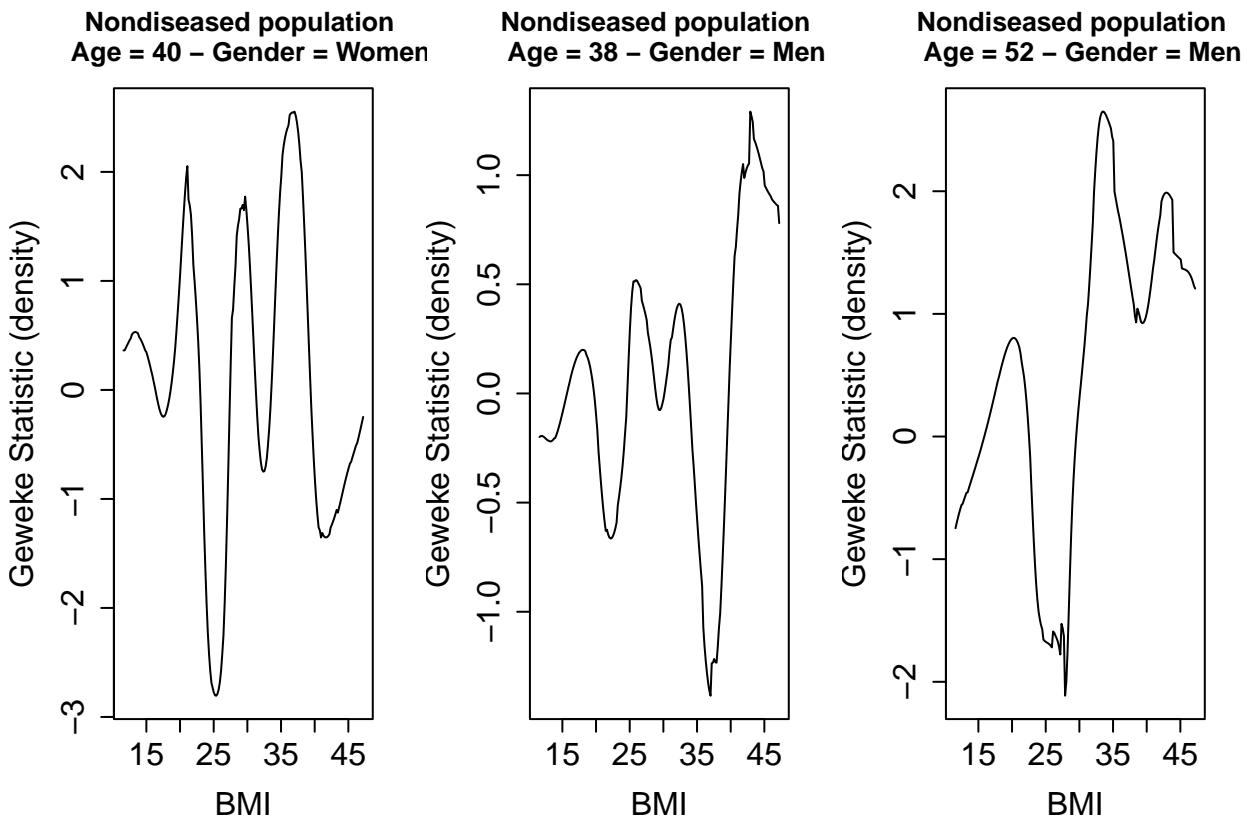
# Diseased
op <- par(mfrow = c(1, 3))
for (i in 1:3) {
  plot(cROC_bnp$dens$d$grid, effectiveSize(t(cROC_bnp$dens$d$dens[,,
    , cov_d_den[i]])), type = "l", xlab = "BMI", ylab = "Effective Sample Size (density)",
    main = paste0("Diseased population \n Age = ", cROC_bnp$newdata$age[cov_d_den[i]],
      " - Gender = ", cROC_bnp$newdata$gender[cov_d_den[i]]),
    cex.main = 1.2, cex.lab = 1.5, cex.axis = 1.5)
}

```



```
par(op)
```

```
# Geweke statistics for the same covariate values as the
# ESS Nondiseased
op <- par(mfrow = c(1, 3))
for (i in 1:3) {
  plot(cROC_bnp$dens$h$grid, geweke.diag(t(cROC_bnp$dens$h$dens[,
    , cov_h_den[i]]))$z, type = "l", xlab = "BMI", ylab = "Geweke Statistic (density)",
    main = paste0("Nondiseased population \n Age = ", cROC_bnp$newdata$age[cov_h_den[i]],
    " - Gender = ", cROC_bnp$newdata$gender[cov_h_den[i]]),
    cex.main = 1.2, cex.lab = 1.5, cex.axis = 1.5)
}
```

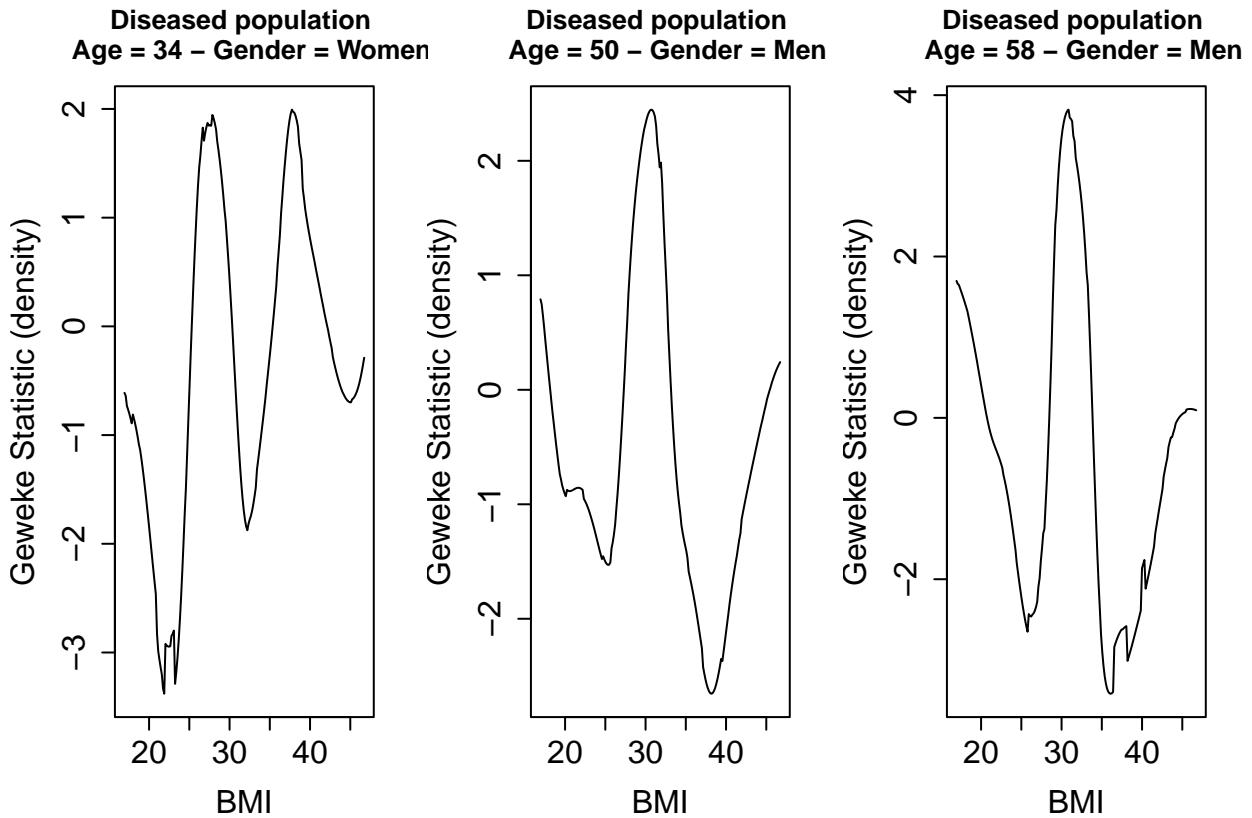


```

par(op)

# Diseased
op <- par(mfrow = c(1, 3))
for (i in 1:3) {
  plot(cROC_bnp$dens$d$grid, geweke.diag(t(cROC_bnp$dens$d$dens[,,
    , cov_d_den[i]]))$z, type = "l", xlab = "BMI", ylab = "Geweke Statistic (density)",
    main = paste0("Diseased population \n Age = ", cROC_bnp$newdata$age[cov_d_den[i]],
      " - Gender = ", cROC_bnp$newdata$gender[cov_d_den[i]]),
    cex.main = 1.2, cex.lab = 1.5, cex.axis = 1.5)
}

```



```
par(op)

# Conditional histograms and densities cROC_bnp object
# NOTE: the code is specific for our application but it
# gives the idea on how conditional histograms and
# densities can be obtained
plot.cden.hist <- function(object, group = c("Nondiseased", "Diseased"),
  age, gender, range_hist, xlab = NULL, title = NULL) {
  group <- match.arg(group)
  if (group == "Nondiseased") {
    denm <- apply(object$dens$h$dens, c(1, 3), mean)
    denl <- apply(object$dens$h$dens, c(1, 3), quantile,
      0.025)
    denh <- apply(object$dens$h$dens, c(1, 3), quantile,
      0.975)
    grid <- object$dens$h$grid
    group.ind <- 0
    title.aux <- "Nondiseased Population"
  } else {
    denm <- apply(object$dens$d$dens, c(1, 3), mean)
    denl <- apply(object$dens$d$dens, c(1, 3), quantile,
      0.025)
    denh <- apply(object$dens$d$dens, c(1, 3), quantile,
      0.975)
    grid <- object$dens$d$grid
    group.ind <- 1
    title.aux <- "Diseased Population"
  }
  plot(denm, grid, type = "h", lwd = 2, col = "black",
    xlab = xlab, ylab = "Density", main = title)
  plot(denl, grid, type = "l", lwd = 1, col = "black",
    xlab = xlab, ylab = "Density", main = title)
  plot(denh, grid, type = "l", lwd = 1, col = "black",
    xlab = xlab, ylab = "Density", main = title)
  grid <- grid[1:(length(grid) - 1)]
  grid[1] <- 0
  grid <- grid[order(grid)]
  axis(1, grid, labels = grid)
  axis(2, 0, 4, labels = c(0, 1, 2, 3, 4))
  if (group.ind == 0) {
    title("Conditional Density Plot for Nondiseased Population")
  } else {
    title("Conditional Density Plot for Diseased Population")
  }
}
```

```

}

pos <- object$newdata$age == age & object$newdata$gender ==
      gender
if (length(range_hist) == 2) {
  pos_data <- object$data$age >= range_hist[1] & object$data$age <
    range_hist[2] & object$data$gender == gender & object$data[,,
    object$group] == group.ind
  xlab.aux <- paste0("BMI (", range_hist[1], " <= Age < ",
    range_hist[2], ")")
} else {
  if (range_hist > age) {
    pos_data <- object$data$age < range_hist & object$data$gender ==
      gender & object$data[, object$group] == group.ind
    xlab.aux <- paste0("BMI (Age < ", range_hist, ")")
  } else {
    pos_data <- object$data$age > range_hist & object$data$gender ==
      gender & object$data[, object$group] == group.ind
    xlab.aux <- paste0("BMI (Age > ", range_hist, ")")
  }
}

df_dens <- data.frame(dm = denm[, pos], dl = denl[, pos],
                      dh = denh[, pos], seqbmi = grid)
df_hist <- data.frame(var1 = object$data[pos_data, object$marker])

if (is.null(title)) {
  title <- paste0(title.aux, ", Gender: ", gender, ", Age: ",
    age)
}
if (is.null(xlab)) {
  xlab <- xlab.aux
}

g <- ggplot(df_dens, aes(x = seqbmi, y = dm)) + geom_line(size = 2) +
  geom_ribbon(data = df_dens, aes(x = seqbmi, ymin = dl,
    ymax = dh), alpha = 0.3, fill = "dodgerblue1") +
  theme_bw() + xlab(xlab) + ylab("Density") + geom_histogram(data = df_hist,
  aes(x = var1, y = after_stat(density)), alpha = 0.2,
  bins = 25, inherit.aes = FALSE) + theme(axis.text = element_text(size = 15),
  axis.title.x = element_text(size = 20), axis.title.y = element_text(size = 20)) +
  ggtitle(title) + theme(plot.title = element_text(), size = 20,
  hjust = 0.5)

g
}

g1 <- plot.cden.hist(cROC_bnp, group = "Nondiseased", age = 30,
                      gender = "Women", range_hist = 35)
g2 <- plot.cden.hist(cROC_bnp, group = "Nondiseased", age = 40,
                      gender = "Women", range_hist = c(35, 45))
g3 <- plot.cden.hist(cROC_bnp, group = "Nondiseased", age = 50,
                      gender = "Women", range_hist = 45)

```

```

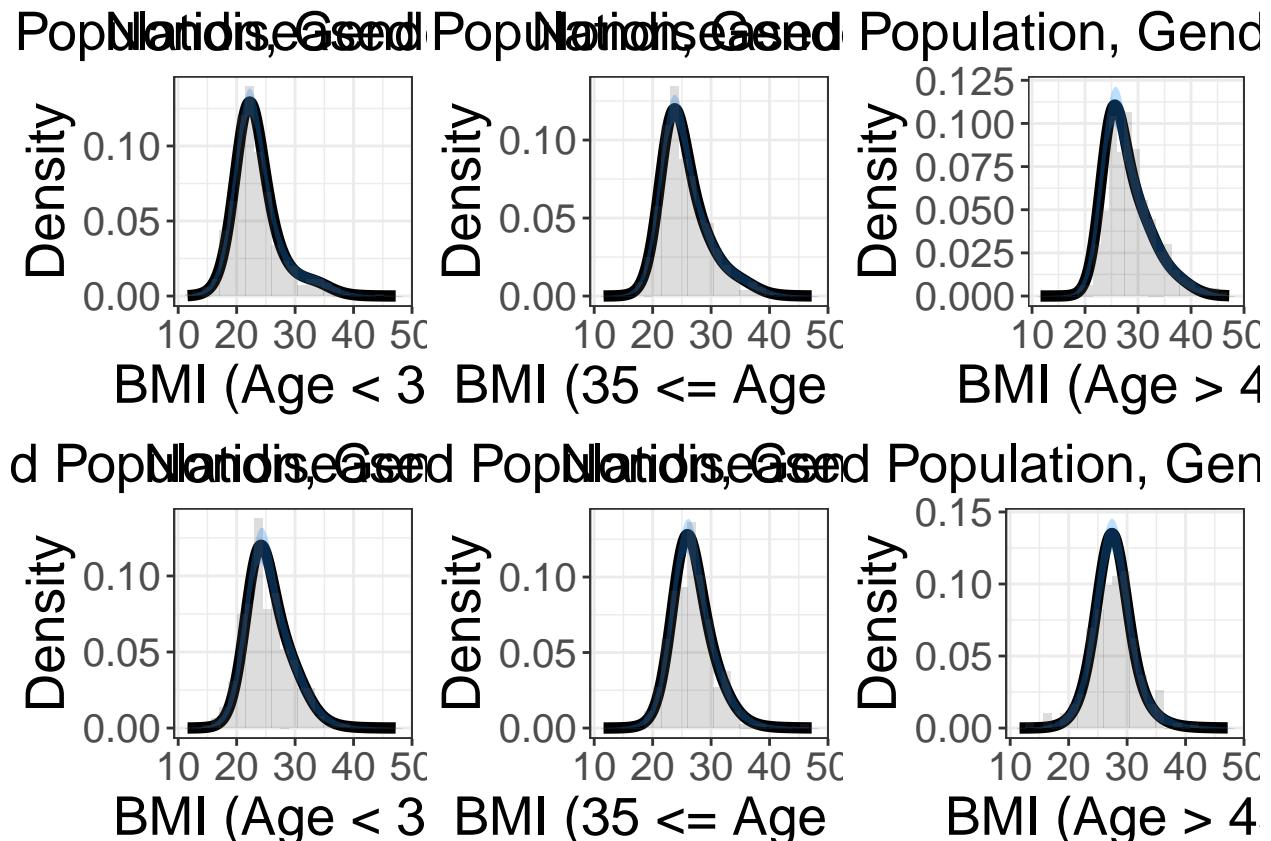
g4 <- plot.cden.hist(cROC_bnp, group = "Nondiseased", age = 30,
                      gender = "Men", range_hist = 35)
g5 <- plot.cden.hist(cROC_bnp, group = "Nondiseased", age = 40,
                      gender = "Men", range_hist = c(35, 45))
g6 <- plot.cden.hist(cROC_bnp, group = "Nondiseased", age = 50,
                      gender = "Men", range_hist = 45)

g7 <- plot.cden.hist(cROC_bnp, group = "Diseased", age = 30,
                      gender = "Women", range_hist = 35)
g8 <- plot.cden.hist(cROC_bnp, group = "Diseased", age = 40,
                      gender = "Women", range_hist = c(35, 45))
g9 <- plot.cden.hist(cROC_bnp, group = "Diseased", age = 50,
                      gender = "Women", range_hist = 45)

g10 <- plot.cden.hist(cROC_bnp, group = "Diseased", age = 30,
                        gender = "Men", range_hist = 35)
g11 <- plot.cden.hist(cROC_bnp, group = "Diseased", age = 40,
                        gender = "Men", range_hist = c(35, 45))
g12 <- plot.cden.hist(cROC_bnp, group = "Diseased", age = 50,
                        gender = "Men", range_hist = 45)

# Nondiseased
grid.arrange(g1, g2, g3, g4, g5, g6, ncol = 3)

```



```
# Deseased  
grid.arrange(g7, g8, g9, g10, g11, g12, ncol = 3)
```

