Lab 1

# Uzduotis NR 1

# 1.Realizuokite dvimačio Gauso atsitiktinių dydžių generavimo funkciją, kuri generuotų dydžius su vidurkiu M ir kovariacine matrica R (žr. variantų lentelę). Ši funkcija gali naudotis tik vienamačio standartinio Gauso atsitiktinio dydžio generatoriumi rnorm(n). Tolimesnei užduočiai naudokite šią savo sudarytą funkciją.  
  
# 6 variantas - Vytautas Kraujalis  
set.seed(6)  
  
# Duoti vidurkiai  
mu1 <- 9  
mu2 <- 1  
  
M <- c(mu1, mu2)  
  
# Duota sigma matrica  
R <- matrix(  
 c(  
 40, -10,  
 -10, 4  
 ),  
 nrow = 2, ncol = 2  
)  
  
# -- 1 --  
  
# Funkcija, kurios argumentai: N - imties dydis, M - Vidurkiu matrica, R - Kovariacine matrica. Funkcija grazina dvimati normaluji a.d. matricos pavidalus 2\*N  
DvimatisGausoAD <- function(N, M, R) {  
 # Cholesky dekompozicija  
  
 # Transponuota cholesky dekompozicija  
 Chol <- t(chol(R))  
  
 # Generuojam normaliuosius vienmacius a.d.  
 Z <- matrix(  
 rnorm(2 \* N),  
 nrow = 2, ncol = N  
 )  
 DvimatisGauso <- t(Chol %\*% Z) +  
 matrix(  
 rep(M, N),  
 byrow = T, ncol = 2  
 )  
 return(DvimatisGauso)  
}  
  
# -------  
# -- 2 --  
  
# Imties dydziu vektorius, su kuriais darysime bandymus  
N <- c(10, 100, 1000, 10000)  
  
print("Tikra kovariacine matrica:")

## [1] "Tikra kovariacine matrica:"

R

## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4

for (i in 1:length(N)) {  
 print(paste0("Kovariacine matrica, kai N = ", N[i]))  
 print(round(cov(DvimatisGausoAD(N[i], M, R)), 1))  
}

## [1] "Kovariacine matrica, kai N = 10"  
## [,1] [,2]  
## [1,] 37.1 -13.4  
## [2,] -13.4 6.5  
## [1] "Kovariacine matrica, kai N = 100"  
## [,1] [,2]  
## [1,] 38.4 -9.2  
## [2,] -9.2 3.6  
## [1] "Kovariacine matrica, kai N = 1000"  
## [,1] [,2]  
## [1,] 38.6 -9.4  
## [2,] -9.4 3.9  
## [1] "Kovariacine matrica, kai N = 10000"  
## [,1] [,2]  
## [1,] 39.9 -9.9  
## [2,] -9.9 4.0

print("Tikri vidurkiai:")

## [1] "Tikri vidurkiai:"

M

## [1] 9 1

for (i in 1:length(N)) {  
 print(paste0("Vidurkiai, kai N = ", N[i]))  
 print(round(colMeans(DvimatisGausoAD(N[i], M, R)), 1))  
}

## [1] "Vidurkiai, kai N = 10"  
## [1] 10.7 0.5  
## [1] "Vidurkiai, kai N = 100"  
## [1] 9.4 1.1  
## [1] "Vidurkiai, kai N = 1000"  
## [1] 9.2 1.0  
## [1] "Vidurkiai, kai N = 10000"  
## [1] 9 1

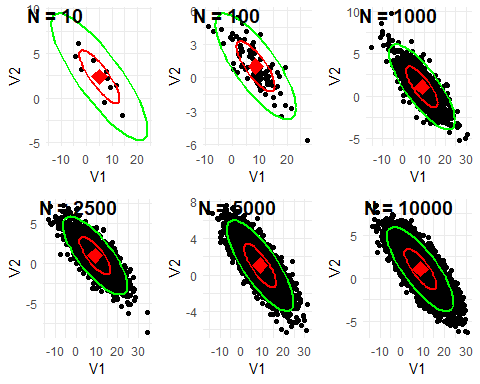
# -------  
# -- 3 --  
  
library(ggplot2)  
library(dplyr)

##   
## Pridedamas paketas: 'dplyr'

## Šie objektai yra užmaskuoti nuo 'package:stats':  
##   
## filter, lag

## Šie objektai yra užmaskuoti nuo 'package:base':  
##   
## intersect, setdiff, setequal, union

# Grafiko brezimas  
BreztiSklaidosDiagrama <- function(DvimatisGausas) {  
 DvimatisGausas <- as.data.frame(DvimatisGausas)  
 # Apskaiciuojam empirinius vidurkius, naudosime tasko padejimui sklaidos diagramoje  
 EmpiriniaiVidurkiai <- DvimatisGausas %>%  
 summarise\_all(mean)  
 # Breziame a.d. sklaidos diagrame, elipsines kreives 95% ir 50% tiketinoms reiksmems ir empirinio vidurkio vieta  
 ggplot(DvimatisGausas, aes(x = V1, y = V2)) +  
 geom\_point() +  
 stat\_ellipse(level = 0.5, color = "red", type = "norm", size = 1) +  
 stat\_ellipse(level = 0.95, color = "green", type = "norm", size = 1) +  
 geom\_point(data = EmpiriniaiVidurkiai, size = 4, color = "red", shape = 23, fill = "red") +  
 theme\_minimal()  
}  
  
# Kuomet keiciam imties dydi N  
N <- c(10, 100, 1000, 2500, 5000, 10000)  
DvimaciaiGauso <- list(  
 DvimatisGausoAD(N[1], M, R),  
 DvimatisGausoAD(N[2], M, R),  
 DvimatisGausoAD(N[3], M, R),  
 DvimatisGausoAD(N[4], M, R),  
 DvimatisGausoAD(N[5], M, R),  
 DvimatisGausoAD(N[6], M, R)  
)  
Labels <- list()  
for (i in 1:length(N)) {  
 Labels[[i]] <- paste0("N = ", N[i])  
}  
  
cowplot::plot\_grid(  
 BreztiSklaidosDiagrama(DvimatisGausoAD(N[1], M, R)),  
 BreztiSklaidosDiagrama(DvimatisGausoAD(N[2], M, R)),  
 BreztiSklaidosDiagrama(DvimatisGausoAD(N[3], M, R)),  
 BreztiSklaidosDiagrama(DvimatisGausoAD(N[4], M, R)),  
 BreztiSklaidosDiagrama(DvimatisGausoAD(N[5], M, R)),  
 BreztiSklaidosDiagrama(DvimatisGausoAD(N[6], M, R)),  
 labels = Labels  
)



ggsave(paste0("Lab1\_Grafikas\_PagalN.png"), device = "png", width = 14, height = 7)  
  
# Funkcija keisti parametrus ir brezti sklaidos diagrama  
# PirmaKoord ir AntraKoord - koordinates, kuriu parametrus keisime; ReiksmeNuo - Koordinates reikmes, kuri bus pradiniu momentu; ReiksmesDidinimas - keliais vienetais didinsime koordinates reiksme; KiekReiksmiu - kiek susigeneruosime reiksmiu (grafikas nubres max 6); N - Imties dydis N  
KeiciamParametrus <- function(PirmaKoord, AntraKoord, ReiksmeNuo, ReiksmesDidinimas, KiekReiksmiu, N) {  
 DvimaciaiGauso <- list()  
 # Kovariacine matrica, kuri kiekvienos iteracijos metu igys nauja parametro reiksme  
 KovariacineMatrica <- R  
 # Isisaugosime koreliacijos koeficientus  
 Koreliacijos <- NULL  
 # Koordinates parametru reiksmes, kurias testuosime  
 koordinate <- seq(ReiksmeNuo, by = ReiksmesDidinimas, length.out = KiekReiksmiu)  
 for (i in 1:KiekReiksmiu) {  
 KovariacineMatrica[PirmaKoord, AntraKoord] <- koordinate[i]  
 DvimaciaiGauso[[i]] <- DvimatisGausoAD(N, M, KovariacineMatrica)  
 print(KovariacineMatrica)  
 # Patikriname, ar tikrines reiksmes yra >= 0  
 print(eigen(KovariacineMatrica)$values >= 0)  
 # Apskaiciuojame ir issisaugome koreliacijos koeficienta  
 Koreliacijos <- c(Koreliacijos, round(cov(DvimaciaiGauso[[i]][, 1], DvimaciaiGauso[[i]][, 2]) / (sqrt(KovariacineMatrica[1, 1]) \* sqrt(KovariacineMatrica[2, 2])), 3))  
 print(Koreliacijos[i])  
 }  
 Labels <- list()  
 for (i in 1:KiekReiksmiu) {  
 Labels[[i]] <- paste0("N = ", N, " Koord. ", PirmaKoord, ", ", AntraKoord, " reiksme = ", koordinate[i], ", kor. koef = ", Koreliacijos[i])  
 }  
 cowplot::plot\_grid(  
 BreztiSklaidosDiagrama(DvimaciaiGauso[[1]]),  
 BreztiSklaidosDiagrama(DvimaciaiGauso[[2]]),  
 BreztiSklaidosDiagrama(DvimaciaiGauso[[3]]),  
 BreztiSklaidosDiagrama(DvimaciaiGauso[[4]]),  
 BreztiSklaidosDiagrama(DvimaciaiGauso[[5]]),  
 BreztiSklaidosDiagrama(DvimaciaiGauso[[6]]),  
 labels = Labels,  
 label\_size = 12,  
 label\_x = -0.3  
 )  
 ggsave(paste0("Lab1\_Grafikas\_", paste(PirmaKoord, AntraKoord, ReiksmeNuo, ReiksmesDidinimas, KiekReiksmiu, sep = "\_"), ".png"), device = "png", width = 14, height = 7)  
}  
  
# Kuomet didinam kovariacines matricos 1,1 koordinate  
KeiciamParametrus(1, 1, 40, 10, 6, 1000)

## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.818  
## [,1] [,2]  
## [1,] 50 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.677  
## [,1] [,2]  
## [1,] 60 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.549  
## [,1] [,2]  
## [1,] 70 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.53  
## [,1] [,2]  
## [1,] 80 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.562  
## [,1] [,2]  
## [1,] 90 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.571

# Kuomet mazinam kovariacines matricos 1,1 koordinate  
KeiciamParametrus(1, 1, 40, -2, 6, 1000)

## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.763  
## [,1] [,2]  
## [1,] 38 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.854  
## [,1] [,2]  
## [1,] 36 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.766  
## [,1] [,2]  
## [1,] 34 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.92  
## [,1] [,2]  
## [1,] 32 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.814  
## [,1] [,2]  
## [1,] 30 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.913

# Kuomet didinam kovariacines matricos 2,2 koordinate  
KeiciamParametrus(2, 2, 10, 50, 6, 1000)

## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 10  
## [1] TRUE TRUE  
## [1] -0.544  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 60  
## [1] TRUE TRUE  
## [1] -0.153  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 110  
## [1] TRUE TRUE  
## [1] -0.128  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 160  
## [1] TRUE TRUE  
## [1] -0.107  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 210  
## [1] TRUE TRUE  
## [1] -0.119  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 260  
## [1] TRUE TRUE  
## [1] -0.05

# Kuomet mazinam kovariacines matricos 2,2 koordinate  
KeiciamParametrus(2, 2, 14, -2, 6, 1000)

## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 14  
## [1] TRUE TRUE  
## [1] -0.413  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 12  
## [1] TRUE TRUE  
## [1] -0.454  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 10  
## [1] TRUE TRUE  
## [1] -0.49  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 8  
## [1] TRUE TRUE  
## [1] -0.541  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 6  
## [1] TRUE TRUE  
## [1] -0.661  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.754

# Kuomet didinam kovariacines matricos 1,2 koordinate  
KeiciamParametrus(1, 2, -10, 4, 6, 1000)

## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.747  
## [,1] [,2]  
## [1,] 40 -6  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.464  
## [,1] [,2]  
## [1,] 40 -2  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.166  
## [,1] [,2]  
## [1,] 40 2  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] 0.084  
## [,1] [,2]  
## [1,] 40 6  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] 0.506  
## [,1] [,2]  
## [1,] 40 10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] 0.752

# Kuomet mazinam kovariacines matricos 1,2 koordinate  
KeiciamParametrus(1, 2, -7, -1, 6, 1000)

## [,1] [,2]  
## [1,] 40 -7  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.545  
## [,1] [,2]  
## [1,] 40 -8  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.67  
## [,1] [,2]  
## [1,] 40 -9  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.712  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.809  
## [,1] [,2]  
## [1,] 40 -11  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.867  
## [,1] [,2]  
## [1,] 40 -12  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.959

# Kuomet didinam kovariacines matricos 2,1 koordinate  
KeiciamParametrus(2, 1, -10, 4, 6, 1000)

## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.753  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -6 4  
## [1] TRUE TRUE  
## [1] -0.742  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -2 4  
## [1] TRUE TRUE  
## [1] -0.752  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] 2 4  
## [1] TRUE TRUE  
## [1] -0.801  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] 6 4  
## [1] TRUE TRUE  
## [1] -0.852  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] 10 4  
## [1] TRUE TRUE  
## [1] -0.807

# Kuomet mazinam kovariacines matricos 2,1 koordinate  
KeiciamParametrus(2, 1, -7, -1, 6, 1000)

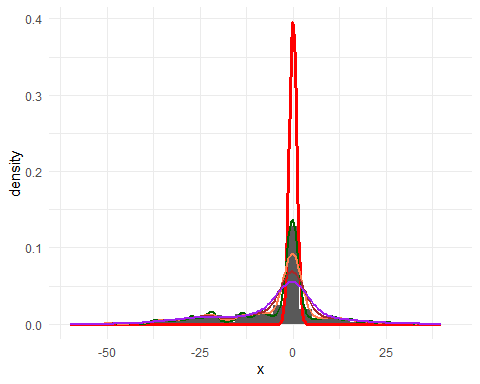
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -7 4  
## [1] TRUE TRUE  
## [1] -0.746  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -8 4  
## [1] TRUE TRUE  
## [1] -0.839  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -9 4  
## [1] TRUE TRUE  
## [1] -0.834  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -10 4  
## [1] TRUE TRUE  
## [1] -0.845  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -11 4  
## [1] TRUE TRUE  
## [1] -0.775  
## [,1] [,2]  
## [1,] 40 -10  
## [2,] -12 4  
## [1] TRUE TRUE  
## [1] -0.715

# -------  
  
# Literatura  
# https://www.itl.nist.gov/div898/handbook/pmc/section5/pmc542.htm  
# https://blog.revolutionanalytics.com/2016/08/simulating-form-the-bivariate-normal-distribution-in-r-1.html  
# https://www2.stat.duke.edu/courses/Spring12/sta104.1/Lectures/Lec22.pdf

# Uzduotis NR 2

# -- 1 --  
q <- 2  
p1 <- 0.5  
mu1 <- 0  
o1 <- 1  
p2 <- 1 - p1  
mu2 <- -8  
o2 <- 16  
  
N <- 300  
U <- runif(N)  
x <- rep(NA, N)  
  
# Susikuriame gauso a.d. mišinį  
for (i in 1:N) {  
 if (U[i] < p1) {  
 x[i] <- rnorm(1, mu1, o1)  
 } else {  
 x[i] <- rnorm(1, mu2, o2)  
 }  
}  
  
# Teorinio tankio funkcijai susigeneruojam x reikšmes  
x\_teorinis <- seq(-60, 40, length.out = 300)  
# Teorinis tankis  
tankis\_teorinis <- p1 \* dnorm(x\_teorinis, mu1, o1) + (1 - p1) \* dnorm(x\_teorinis, mu1, o1)  
  
df <- data.frame(x = x, tankis\_teorinis = tankis\_teorinis, x\_teorinis = x\_teorinis)  
  
ggplot(df) +  
 geom\_histogram(aes(x = x, y = ..density..)) +  
 geom\_line(aes(x = x\_teorinis, y = tankis\_teorinis), color = "red", size = 1.4) +  
 stat\_density(aes(x = x), geom = "line", kernel = "gaussian", n = 300, adjust = 0.5, color = "darkgreen", size = 1) +  
 stat\_density(aes(x = x), geom = "line", kernel = "gaussian", n = 300, adjust = 1, color = "coral", size = 1) +  
 stat\_density(aes(x = x), geom = "line", kernel = "gaussian", n = 300, adjust = 1.5, color = "brown", size = 1) +  
 stat\_density(aes(x = x), geom = "line", kernel = "gaussian", n = 300, adjust = 2, color = "blue", size = 1) +  
 stat\_density(aes(x = x), geom = "line", kernel = "gaussian", n = 300, adjust = 2, color = "purple", size = 1) +  
 theme\_minimal()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggsave("Lab1\_Grafikas\_AD\_Branduoliai\_.png", device = "png", width = 14, height = 7)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

# -------  
# -- 2 --  
q <- 2  
p1 <- 0.2  
M1 <- c(0, 0)  
R1 <- matrix(  
 c(  
 1, 0,  
 0, 1  
 ),  
 ncol = 2, nrow = 2  
)  
M2 <- c(-15, 15)  
R2 <- matrix(  
 c(  
 10, -10,  
 -10, 30  
 ),  
 ncol = 2, nrow = 2  
)  
  
# Mišinio vidurkiai  
M <- matrix(c(M1, M2), ncol = 2, byrow = T)  
  
# Mišinio kov. matricos  
R <- array(rep(NA, 2 \* 2 \* 2), c(2, 2, 2))  
R[, , 1] <- R1  
R[, , 2] <- R2  
  
library(MASS)

##   
## Pridedamas paketas: 'MASS'

## Šis objektas yra užmaskuotas nuo 'package:dplyr':  
##   
## select

N <- 500  
  
# Dvimačio Gauso A.D. mišinio generavimo funkcija, N - imties dydis; M\_2d - Mišinio vidurkiai; R\_2d - Mišinio kov. matricos  
DvimatisGausoAD\_Misinys <- function(N, M\_2d, R\_2d) {  
 matrica <- matrix(rep(NA, N \* 2), ncol = 2)  
 colnames(matrica) <- paste0("X", 1:2)  
 U <- runif(N)  
 for (i in 1:N) {  
 if (U[i] < p1) {  
 matrica[i, ] <- DvimatisGausoAD(1, M = M\_2d[1, ], R = R\_2d[, , 1])  
 } else {  
 matrica[i, ] <- DvimatisGausoAD(1, M = M\_2d[2, ], R = R\_2d[, , 2])  
 }  
 }  
 return(matrica)  
}  
  
dvimatis\_gauso\_ad\_misinys <- DvimatisGausoAD\_Misinys(N, M, R)  
  
png("dvimatis\_gauso\_ad\_misinys\_sklaidosdiagrama.png", width = 14, height = 7, units = "in", res = 1200)  
plot(dvimatis\_gauso\_ad\_misinys)  
dev.off()

## png   
## 2

BreztiDvimacioGausoAD\_Misinio\_Tanki <- function(H) {  
 kde <- ks::kde(x = dvimatis\_gauso\_ad\_misinys, H = H)  
 png(paste0("heatmap\_dvimatisAD\_", H[1, 1], "\_", H[1, 2], "\_", H[2, 1], "\_", H[2, 2], ".png"), width = 14, height = 7, units = "in", res = 1200)  
 image(kde$eval.points[[1]], kde$eval.points[[2]], kde$estimate,  
 col = viridis::viridis(20), xlab = "X1", ylab = "X2"  
 )  
 points(kde$x)  
 dev.off()  
  
 png(paste0("3d\_dvimatisAD\_", H[1, 1], "\_", H[1, 2], "\_", H[2, 1], "\_", H[2, 2], ".png"), width = 14, height = 7, units = "in", res = 1200)  
 plot(kde, display = "persp", col.fun = viridis::viridis, xlab = "X1", ylab = "X2")  
 dev.off()  
}  
  
# Keičiam [1, 1] reikšmę  
H1 <- matrix(c(2, -1.4, -1.4, 1), ncol = 2, nrow = 2)  
H2 <- matrix(c(5, -1.4, -1.4, 1), ncol = 2, nrow = 2)  
H3 <- matrix(c(10, -1.4, -1.4, 1), ncol = 2, nrow = 2)  
  
BreztiDvimacioGausoAD\_Misinio\_Tanki(H1)

## png   
## 2

BreztiDvimacioGausoAD\_Misinio\_Tanki(H2)

## png   
## 2

BreztiDvimacioGausoAD\_Misinio\_Tanki(H3)

## png   
## 2

# Keičiam [2, 2] reikšmę  
H5 <- matrix(c(1.7, -1.4, -1.4, 1.5), ncol = 2, nrow = 2)  
H6 <- matrix(c(1.7, -1.4, -1.4, 5), ncol = 2, nrow = 2)  
H7 <- matrix(c(1.7, -1.4, -1.4, 10), ncol = 2, nrow = 2)  
  
BreztiDvimacioGausoAD\_Misinio\_Tanki(H5)

## png   
## 2

BreztiDvimacioGausoAD\_Misinio\_Tanki(H6)

## png   
## 2

BreztiDvimacioGausoAD\_Misinio\_Tanki(H7)

## png   
## 2

# Keičiam [1, 2] ir [2, 1] reikšmę  
H8 <- matrix(c(1.7, -1.4, -1.4, 2.4), ncol = 2, nrow = 2)  
H9 <- matrix(c(1.7, 0, 0, 2.4), ncol = 2, nrow = 2)  
H10 <- matrix(c(1.7, 2, 2, 2.4), ncol = 2, nrow = 2)  
  
BreztiDvimacioGausoAD\_Misinio\_Tanki(H8)

## png   
## 2

BreztiDvimacioGausoAD\_Misinio\_Tanki(H9)

## png   
## 2

BreztiDvimacioGausoAD\_Misinio\_Tanki(H10)

## png   
## 2

# -------  
  
# Literatura  
# https://stats.stackexchange.com/questions/70855/generating-random-variables-from-a-mixture-of-normal-distributions  
# http://www.di.fc.ul.pt/~jpn/r/EM/GaussianMix.html  
# https://bookdown.org/egarpor/NP-UC3M/kde-ii-mult.html  
# https://mfasiolo.github.io/sc2-2019/rcpp\_advanced\_i/3\_kde\_armadillo/