**TUGAS ANALISIS PEUBAH GANDA**

**VISUALISASI DISTRIBUSI NORMAL BIVARIATE**

**dengan TCLTK**

**Disusun oleh:**

**TINGKAT 4 KOMPUTASI STATISTIK**

**KELOMPOK 4**

**TIM VISUALISASI TCLTK**

**SEKOLAH TINGGI ILMU STATISTIK**

**2015/2016**

**Anggota Kelompok**

1. Achmad Fauzi Bagus Firmansyah 4KS1 / 12.6980
2. Dwi Puspita Sari 4KS1 / 12.7116
3. Immawan Mujahid Maruddani 4KS1 / 12.7183
4. Khairanisa Haque 4KS1 / 12.7210
5. Kurnia Fauzi 4KS1 / 12.7215
6. M. Arif Rosyanto 4KS1 / 12.7232
7. Nur Azizah 4KS1 / 12.7291
8. Ristika Nugraha 4KS1 / 12.7350
9. Shintia Hafifah 4KS1 / 12.7383
10. Choerul Afifanto 4KS2 / 12.7077
11. Eko Wahyu Lestari 4KS2 / 12.7121
12. M. Abdul Muhshi 4KS2 / 12.7230
13. Moh Syaiful Hidayatur Rakhman 4KS2 / 12.7253
14. Muhammad Fikri Firmansyah 4KS2 / 12.7263
15. Tegar Dani Pratama 4KS2 / 12.7401

|  |  |  |
| --- | --- | --- |
| **Fungsi untuk men-generate data bivariate normal** | | |
| **Membuat fungsi dg nilai default parameternya**  **dan** | Generate.data <- function (mu1=0, mu2=0, sig1=0.35, sig2=0.35, rho=0.5) { | |
| **Inisialisasi untuk x dan y** | xm <- -3  xp <- 3  ym <- -3  yp <- 3  x <- seq(xm, xp,  length = as.integer((xp+abs(xm))\*10))  y <- seq(ym, yp,  length = as.integer((xp+abs(ym))\*10)) | |
| **Fungsi kepadatan peluang dari bivariate normal** | | |
| **Fungsi peluang kepadatan dari bivariate normal** | | bivariate <- function(x,y){  term1 <- 1 / (2 \* pi \* sig1 \* sig2 \*  sqrt(1 - rho^2))  term2 <- (x-mu1)^2 / sig1^2  term3 <- -(2 \* rho \* (x - mu1)\*  (y - mu2))/(sig1\*sig2)  term4 <- (y - mu2)^2 / sig2^2  z <- term2 + term3 + term4  term5 <- term1 \* exp((-z/(2\*(1-rho^2))))  return(term5)  } |
| **Untuk mendapatkan matriks nilai kepadatan peluang bivariate normal dari x dan y** | | z <- outer(x,y,bivariate) |
| **return list nilai**  **Array x, array y, dan z untuk matriks kepadatan peluang bivariate normal dari x dan y** | | return(list(mu1 = mu1, mu2 = mu2,  sig1 = sig1, sig2 = sig2,  rho = rho,x = x, y = x, z = z)) |
| **Menutup fungsi** | | } |

|  |  |
| --- | --- |
| **Fungsi Plot Perspective** | |
| **Membuat fungsi bernama BiVPers** | BiVPers <- function(generate.data){ |
| **Memberi nilai untuk variabel x, y, z, mu1, mu2, sig1, sig2, rho** | z<-generate.data$z  x<-generate.data$x  y<-generate.data$y  mu1<-generate.data$mu1  mu2<-generate.data$mu2  sig1<-generate.data$sig1  sig2<-generate.data$sig2  rho<-generate.data$rho |
| **Membuat plot perspective** | co <- c('cyan','blue')  f <- facetcol(co,z)  persp(x, y, z,  sub = bquote(bold(mu[1])==.(mu1) ~", "~  sigma[1]==.(sig1) ~", "~ mu[2]==.(mu2)  ~", "~ sigma[2]==.(sig2) ~", "~  rho==.(rho)),  col=color[facetcl], theta = 55, phi = 30,  r = 40, d = 0.1, expand = .9,  ltheta = 90, lphi = 180, shade = 0.4,  ticktype = "detailed", nticks=5) |
| **Menutup fungsi** | } |

|  |  |
| --- | --- |
| **Fungsi untuk membuat plot contour dari data distribusi normal bivariat** | |
| **Membuat fungsi bernama BiVContour** | BiVContour <- function(generate.data){ |
| **Memberi nilai untuk variabel x, y, z, mu1, mu2, sig1, sig2, rho** | z<-generate.data$z  x<-generate.data$x  y<-generate.data$y  mu1<-generate.data$mu1  mu2<-generate.data$mu2  sig1<-generate.data$sig1  sig2<-generate.data$sig2  rho<-generate.data$rho |
| **Membuat contour** | co <- c('white','purple')  f <- facetcol(co,z)  image(x, y, z,  col = rainbow(15, start = 3/6, end =4/6),  sub = bquote(bold(mu[1])==.(mu1) ~", "~  sigma[1]==.(sig1) ~", "~ mu[2]==.(mu2)  ~", "~ sigma[2]==.(sig2) ~", "~  rho==.(rho))  )  contour(x, y, z, nlevels = 15,  drawlabels = TRUE,  method = "flattest",  axes = TRUE,  col = color[facetcl],  lty = par("lty"), lwd=par("lwd"), add=T  ) |
| **Menutup fungsi** | } |

|  |  |
| --- | --- |
| **Dokumentasi GUI.R** | |
| **Fungsi ini membutuhkan package tcltk, tcltk2, tkrplot** | require (tcltk)  require (tcltk2)  require (tkrplot) |
| **Fungsi ini memanggil file Generate-data.R, Perspective.R, dan Contour.R** | source("Generate-data.R")  source("Perspective.R")  source("Contour.R") |
| **Membuat fungsi bernama BivariateGUI** | BivariateGUI <- function(){ |
| **Membuat Main Window** | tt <- tktoplevel()  tktitle(tt)<-"Bivariate Visualization" |
| **Membuat menu** | tt$env$menu <- tk2menu(tt)  tkconfigure(tt, menu = tt$env$menu) tt$env$menuFile <- tk2menu(tt$env$menu, tearoff = FALSE)  tkadd(tt$env$menuFile, "command", label = "Close", command = function() tkdestroy(tt))  tkadd(tt$env$menu, "cascade", label = "File", menu = tt$env$menuFile)  tt$env$menuAbout <- tk2menu(tt$env$menu, tearoff = FALSE)  tkadd(tt$env$menu, "command", label = "About",  command =function() tkmessageBox(title = "About Program",  message = "Tugas Analisis Peubah Ganda 4KS1-4KS2 Tahun 2016", icon = "info", type = "ok"))    "Arial",size = 16,weight = "bold")  fontTitle<- tkfont.create(size = 16,weight = "bold")    tkgrid(  tk2label(  tt, text = "Bivariate Normal Simulation", justify = "center",  font=fontTitle  ),  padx = 10, pady = c(20, 5),row=0,column=0  ) |
| **Input Varians** | varx <- tclVar("2.1")  vary <- tclVar("2.1")    varians<<-c(round(as.double(tclvalue(varx)),2),  round(as.double(tclvalue(vary)),2)) |
| **Membuat Slider µ** | slidermiuX <- tclVar("0.5")  slidermiuY <- tclVar("0.5")    miu<<-c(round(as.double(tclvalue(slidermiuX)),2),  round(as.double(tclvalue(slidermiuY)),2)) |
| **Input Kovarians** | corxy <- tclVar(".5")  correlation<<-round(as.double(tclvalue(corxy)),2) |
| **Membuat Frame untuk Plotting** | plot.frame<-tk2frame(tt,borderwidth=2,relief="flat", padding=10)  tkgrid(plot.frame,padx=0,pady=c(0,0),row=1,column=0,sticky="w") |
| **Membuat Default Testing Plot** | plotBivaPers <- function() {  generate.data <- Generate.data(mu1=miu[1], mu2=miu[2], sig1=varians[1],  sig2=varians[2], rho=correlation)  BiVPers(generate.data)    }  plotBivaCon <- function() {  generate.data <- Generate.data(mu1=miu[1], mu2=miu[2], sig1=varians[1],  sig2=varians[2], rho=correlation)  BiVContour(generate.data)  }    persp.plot<-tkrplot(plot.frame,plotBivaPers,1.25,1.25)  cont.plot<-tkrplot(plot.frame,plotBivaCon,1.25,1.25)  tkgrid(persp.plot,cont.plot,row=0,padx=5) |
| **Membuat fungsi untuk mengubah plot** | performed<-function(){  tkrreplot(persp.plot)  tkrreplot(cont.plot)  } |
| **Membuat Frame untuk input** | input.frame<-tk2frame(tt,borderwidth=2,relief="flat")  tkgrid(input.frame,padx=20,pady=c(0,20),row=2,column=0,sticky="w")  tkgrid(  tk2label(  input.frame,text="Parameter Value"  #,font=fontCommand  ),  padx= 5,pady=c(5,5),row=0,column=0,sticky="w"  )    labelmiux <- tk2label(input.frame,  text = "miu x= 0.5",justify="left")  tkgrid(labelmiux, padx = 5, pady = c(5, 5),row=1,column=0,sticky="w")  labelmiuY <- tk2label(input.frame,  text = "miu y= 0.5",justify="left")  tkgrid(labelmiuY, padx = 5, pady = c(5, 5),row=1,column=5,sticky="w")    onChangeX <- function(...) {  value <- round(as.double(tclvalue(slidermiuX)),2)  label <- sprintf("miu x= %s", value)  tkconfigure(labelmiux, text = label)  miu<<-c(value,round(as.double(tclvalue(slidermiuY)),2))  performed()    }  onChangeY <- function(...) {  value <- round(as.double(tclvalue(slidermiuY)),2)  label <- sprintf("miu y= %s", value)  tkconfigure(labelmiuY, text = label)  miu<<-c(round(as.double(tclvalue(slidermiuX)),2),  value)  performed()  }    miux.slider <- tk2scale(input.frame,from=-3,to=3,  variable = slidermiuX,  orient = "horizontal", length = 100,  command = onChangeX)  tkgrid(miux.slider, padx = 5, pady = c(5, 5),row=1,column=1,sticky="w")    miuY.slider <- tk2scale(input.frame, from = -3, to =3,  variable = slidermiuY,  orient = "horizontal", length = 100,  command = onChangeY)  tkgrid(miuY.slider, padx = 5, pady = c(5, 5),row=1,column=6,sticky="w")      labelvarx <- tk2label(input.frame,  text = "var x= 2.1",justify="left")  tkgrid(labelvarx, padx = 5, pady = c(5, 5),row=1,column=2,sticky="w")  labelvary <- tk2label(input.frame,  text = "var y= 2.1",justify="left")  tkgrid(labelvary, padx = 5, pady = c(5, 5),row=1,column=7,sticky="w")    onChangeVarX <- function(...) {  value <- round(as.double(tclvalue(varx)),2)  label <- sprintf("var x= %s", value)  tkconfigure(labelvarx, text = label)  varians<<-c(round(as.double(tclvalue(varx)),2),  round(as.double(tclvalue(vary)),2))  performed()  }  onChangeVarY <- function(...) {  value <- round(as.double(tclvalue(vary)),2)  label <- sprintf("var y= %s", value)  tkconfigure(labelvary, text = label)  varians<<-c(round(as.double(tclvalue(varx)),2),  round(as.double(tclvalue(vary)),2))  performed()  }    var.x.slider <- tk2scale(input.frame,from=0.01,to=2.5,  variable = varx,  orient = "horizontal", length = 100,  command = onChangeVarX)  tkgrid(var.x.slider, padx = 5, pady = c(5, 5),row=1,column=3,sticky="w")    var.y.slider <- tk2scale(input.frame, from = 0.01, to =2.5,  variable = vary,  orient = "horizontal", length = 100,  command = onChangeVarY)  tkgrid(var.y.slider, padx = 5, pady = c(5, 5),row=1,column=8,sticky="w")    labelcorxy <- tk2label(input.frame,  text = "cor xy= 0.5",justify="left")  tkgrid(labelcorxy, padx = 5, pady = c(5, 5),row=1,column=9,sticky="w")  onChangeCorXY <- function(...) {  value <- round(as.double(tclvalue(corxy)),2)  label <- sprintf("cor xy= %s", value)  tkconfigure(labelcorxy, text = label)  correlation<<-round(as.double(tclvalue(corxy)),2)  performed()  }  cor.xy.slider <- tk2scale(input.frame,from=-.99,to=.99,  variable = corxy,  orient = "horizontal", length = 100,  command = onChangeCorXY)  tkgrid(cor.xy.slider, padx = 5, pady = c(5, 5),row=1,column=10,sticky="w")  sg <- ttksizegrip(tt)  tkgrid(sg, sticky="es") |
| **Menutup fungsi** | } |
| **Menjalankan BivariateGUI** | BivariateGUI() |

