Raport

Repartitii Discrete si Continue

Mechie Daria-Elena, Rada Paul, Paralescu Flavia-Patricia

02.02.2023

Descrierea aplicatiei

Aceasta este o aplicatie care ilustreaza o serie de repartitii discrete si continue si modul in care pot fi utilizate in calcul: Pentru fiecare repartitie avem posibilitatea de a selecta valorile parametrilor care definesc repartitia de a ilustra grafic densitatea(respectiv functia de masa) si functia de repartitie dar si de a calcula probabilitati de tipul $P(X \leq a)$, $P(X \geq b)$ sau $P(a \leq X \leq b)$, pentru valori ale lui a si b date si se poate ilustra grafic probabilitatea calculata prin colaborarea zonei corespunzatoare.

Constructia aplicatiei

Am inceput prin a alege layout-ul aplicatiei: un panel care contine o selectie de repartitii, doua **numericInput**-uri pentru alegerea parametrilor comuni tuturor repartitiilor (a si b), si unul sau mai multe **numericInput**-uri pentru a introduce valorile parametrilor repartitiei alese. Pentru a selecta repartitia dorita am folosit **selectInput**, iar apoi am reprezentat graficul repartitiei alese. Graficele functiilor, valorile probabilitatilor si graficul colorat sunt afisate prin intermediul a doua **plotOutput**-uri si a trei **textOutput**-uri care sunt aranjate cu ajutorul unui **fluidRow** pe doua coloane.

```
library(shiny)
library(shinyjs)
library(shinydashboard)

ui <- fluidPage(              tags$style('.container-fluid {
                             background-color: bfbfbf;
              '),
  titlePanel("Proiect Probabilitati si Statistica"),

  sidebarLayout(
    position = "right",
    sidebarPanel(

      useShinyjs(),
      selectInput("var",  h3("Alegeti Repartitia"),
                  choices = list("Beta" = 1,
                                 "Binomiala" = 2,
                                 "Cauchy" = 3,
                                 "Chy-Squagray" = 4,
                                 "Exponentiala" = 5,
                                 "Fisher" = 6,
                                 "Gamma" = 7,
```

```
                                "Hipergeometrica" = 8,
                                "Log-Normala" = 9,
                                "Logistica" = 10,
                                "Normala" = 11,
                                "Poisson" = 12,
                                "Student" = 13,
                                "Uniforma" = 14,
                                "Weibull" = 15),
                selected = 1),
    helpText(h3("Parametrii a si b")),
    numericInput( "a",
                  h4("a"),
                  min = 0,
                  value = 0,
                  step = 1),
    numericInput( "b",
                  h4("b"),
                  min = 0,
                  value = 0,
                  step = 1),
    helpText(h3(" ")),
    helpText(h3("Parametrii repartitiei")),
    box(id= "uniforma",  width = '800px',
        numericInput( "minU",
                      h3("min"),
                      value = 0),
        numericInput( "maxU",
                      h3("max"),
                      value = 1)
    ),
    box(id= "binomiala",  width = '800px',
        numericInput( "n",
                      h3("n"),
                      value = 1),
        numericInput( "p",
                      h3("p"),
                      value = 0,
                      min = 0,
                      max = 1,
                      step = 0.1)
    ),
    box(id= "cauchy",  width = '800px',
        numericInput( "t",
                      h3("t"),
                      value = 0),
        numericInput( "s",
                      h3("s"),
                      value = 1,
                      min = 0)
    ),
    box(id= "Chy-Squagray",  width = '800px',
        numericInput( "k",
                      h3("k"),
```

```
                    min = 1,
                    value = 1,
                    step = 1)
    ),
    box(id= "exponentiala",  width = '800px',
        numericInput( "lambda",
                    h3("lambda"),
                    min = 0.1,
                    value = 1,
                    step = 0.1)
    ),
    box(id= "fisher",  width = '800px',
        numericInput( "d1",
                    h3("d1"),
                    min = 0.1,
                    value = 0.1,
                    step = 0.1),
        numericInput( "d2",
                    h3("d2"),
                    min = 0.1,
                    value = 0.1,
                    step = 0.1)
    ),
    box(id= "gamma",  width = '800px',
        numericInput( "alpha",
                    h3("alpha"),
                    min = 0.1,
                    value = 0.1,
                    step = 0.1),
        numericInput( "beta",
                    h3("beta"),
                    min = 0.1,
                    value = 0.1,
                    step = 0.1)
    ),
    box(id= "hipergeometrica",  width = '800px',
        numericInput( "m_",
                    h3("m (numarul de bile albe)"),
                    min = 0,
                    value = 0,
                    step = 1),
        numericInput( "n_",
                    h3("n (numarul de bile negre)"),
                    min = 0,
                    value = 0,
                    step = 1),
        numericInput( "k_",
                    h3("k (numarul de bile extrase)"),
                    min = 0,
                    value = 0,
                    step = 1)
    ),
    box(id= "log-normala",  width = '800px',
```

```
        numericInput( "mu",
                      h3("mu"),
                      value = 0,
                      step = 0.1),
        numericInput( "teta",
                      h3("teta"),
                      min = 0,
                      value = 1,
                      step = 0.1)
    ),
    box(id= "logistica",  width = '800px',
        numericInput( "tt",
                      h3("t"),
                      value = 0),
        numericInput( "ss",
                      h3("s"),
                      value = 1,
                      min = 0)
    ),
    box(id= "normala",  width = '800px',
        numericInput( "mu_",
                      h3("mu"),
                      value = 0,
                      step = 0.1),
        numericInput( "teta_",
                      h3("teta^2"),
                      min = 0,
                      value = 1,
                      step = 0.1)
    ),
    box(id= "poisson",  width = '800px',
        numericInput( "lambda_",
                      h3("lambda"),
                      min = 0.1,
                      value = 0.1,
                      step = 0.1)
    ),
    box(id= "student",  width = '800px',
        numericInput( "d",
                      h3("d"),
                      min = 0,
                      value = 1,
                      step = 1)
    ),
    box(id= "Beta",  width = '800px',
        numericInput( "alpha_",
                      h3("alpha"),
                      min = 0.1,
                      value = 0.1,
                      step = 0.1),
        numericInput( "beta_",
                      h3("beta"),
                      min = 0.1,
```

```
                        value = 0.1,
                        step = 0.1)
      ),
      box(id= "weibull",  width = '800px',
          numericInput( "kk",
                        h3("k"),
                        min = 0.1,
                        value = 1,
                        step = 0.1)
      ),


    ),

    mainPanel(

      plotOutput("functii"),
      fluidRow(
        column(6,
               div(style = "height:20px; font-size:35px; color: pink; margin-bottom: 40px; margin-top:
               div(style = "height:20px; font-size:35px; color: #FF00FF; margin-bottom: 40px; font-weig
               div(style = "height:20px; font-size:35px; color: purple; font-weight: bold;", textOutput
        ),
        column(6,
               plotOutput("Px_a")
        )
      )
    )
  )
)
```

In functie de repartitia aleasa, am afisat valorile specifice. Apoi, am trasat graficele functiei de masa/densitatii si functiei de repartitie. Am calculat probabilitatile de tipul $P(X \leq a)$, $P(X \geq b)$ sau $P(a \leq X \leq b)$, pentru valorile lui a si b date si am ilustrat grafic probabilitatile calculate prin colorarea zonelor corespunzatoare.

```
server <- function(input, output) {

  observeEvent(input$var, {
    # if pentru fiecare repartitie
  })
}
```

Pentru a afisa interfata si a porni server-ul am rulat urmatoarea comanda:

```
shinyApp(ui = ui, server = server)
```

Repartitiile abordate

1. Repartitia Beta

```
    if(input$var == 1)
    { shinyjs::hide(id = "uniforma")
      shinyjs::hide(id = "binomiala")
      shinyjs::hide(id = "cauchy")
      shinyjs::hide(id = "Chy-Squagray")
      shinyjs::hide(id = "exponentiala")
      shinyjs::hide(id = "fisher")
      shinyjs::hide(id = "gamma")
      shinyjs::hide(id = "hipergeometrica")
      shinyjs::hide(id = "log-normala")
      shinyjs::hide(id = "logistica")
      shinyjs::hide(id = "poisson")
      shinyjs::hide(id = "student")
      shinyjs::hide(id = "normala")
      shinyjs::show(id = "Beta")
      shinyjs::hide(id = "weibull")
      output$functii <- renderPlot({par(mfrow = c(1, 2))
        curve(dbeta(x, shape1 = input$alpha_, shape2 = input$beta_), 0, 1, ylim = c(0, 2.5),
              ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
        curve(pbeta(x, shape1 = input$alpha_, shape2 = input$beta_),0 ,1, ylim = c(0, 1), type = "l", ma
              ylab = "F(x)", lwd = 2, col = "gray")})

      output$Px_a_val <- renderText({
        paste("P( X <= a ) = P( X <=", input$a, ") =", round(pbeta(input$a, shape1 = input$alpha_, shape

      })
      output$Px_b_val <- renderText({
        paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pbeta(input$b, shape1 = input$alpha_, s

      })
      output$Px_a_b_val <- renderText({
        paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((pbeta(input$b, shape1 =
      })

      output$Px_a <- renderPlot({curve(pbeta(x, shape1 = input$alpha_, shape2 = input$beta_),0, 1, ylim
                                       ylab = "F(x)", lwd = 2, col = "gray")

        x <- seq(0,input$a, by=0.01)
        x1 <- c(x, seq(input$a,0, by=-0.01))
        y <- c(c(pbeta(x, shape1 = input$alpha_, shape2 = input$beta_)),seq(0,0, length = length(x)))
        polygon(x1, y, col = "pink")

        x <- seq(input$b, 1, by=0.01)
        x1 <- c(x, seq(1, input$b, by=-0.01))
        y <- c(c(pbeta(x,shape1 = input$alpha_, shape2 = input$beta_)),seq(0,0, length = length(x)))
        polygon(x1, y, col = "#FF00FF")

        x<-seq(input$a, input$b, by=0.01)
        x1 <- c(x, seq(input$b, input$a, by=-0.01))
```

```
      y <- c(c(pbeta(x, shape1 = input$alpha_, shape2 = input$beta_)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

2. Repartitia Binomiala

```
  if(input$var == 2)
  {shinyjs::hide(id = "uniforma")
    shinyjs::show(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      plot(seq(0,40,1),dbinom(seq(0,40,1),size = input$n, prob = input$p),type="p", xlab = "x", ylab =
           lwd = 2, col = "gray")
      plot(seq(0,40,1),pbinom(seq(0,40,1), size = input$n, prob = input$p),type="p", xlab = "x", ylab
           main = "Functia de repartitie", lwd = 2, col = "gray")})

    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(pbinom(input$a, size = input$n, prob = inpu

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pbinom(input$b, size = input$n, prob =

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((pbinom(input$b, size = i

    })
    output$Px_a <-renderPlot({plot(seq(0,40,1),pbinom(seq(0,40,1), size = input$n, prob = input$p),typ
                                  main = "Probabilitati", lwd = 2, col = "gray")
      x <- seq(0,input$a, by=0.2)
      x1 <- c(x, seq(input$a,0, by=-0.2))
      y <- c(c(pbinom(x, size = input$n, prob = input$p)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "pink")

      x <- seq(input$b, input$n, by=0.2)
      x1 <- c(x, seq(input$n, input$b, by=-0.2) )
      y <- c(c(pbinom(x, size = input$n, prob = input$p)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "#FF00FF")
```

```
      x<-seq(input$a, input$b, by=0.2)
      x1 <- c(x, seq(input$b, input$a, by=-0.2))
      y <- c(c(pbinom(x, size = input$n, prob = input$p)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

3. Repartitia Cauchy

```
  if(input$var == 3)
  {shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::show(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      curve(dcauchy(x, location  = input$t, scale = input$s), input$t-2, input$t+2, ylim = c(0, dcauch
            ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
      curve(pcauchy(x, location  = input$t, scale = input$s),input$t-2, input$t+2, ylim = c(0, 1), typ
            ylab = "F(x)", lwd = 2, col = "gray")})

    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(pcauchy(input$a, location  = input$t, scale

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pcauchy(input$b, location  = input$t, s

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((pcauchy(input$b, locati
    })

    output$Px_a <- renderPlot({curve(pcauchy(x, location  = input$t, scale = input$s),input$t-2, input
                                     ylab = "F(x)", lwd = 2, col = "gray")

      x <- seq(input$t-2,input$a, by=0.2)
      x1 <- c(x, seq(input$a,input$t-2, by=-0.2))
      y <- c(c(pcauchy(x, location  = input$t, scale = input$s)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "pink")
```

```
      x <- seq(input$b, input$t+2, by=0.2)
      x1 <- c(x, seq(input$t+2, input$b, by=-0.2))
      y <- c(c(pcauchy(x, location  = input$t, scale = input$s)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "#FF00FF")

      x<-seq(input$a, input$b, by=0.2)
      x1 <- c(x, seq(input$b, input$a, by=-0.2))
      y <- c(c(pcauchy(x, location  = input$t, scale = input$s)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

4. Repartitia Chy-Squared

```
  if(input$var == 4)
  {shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::show(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      curve(dchisq(x, df  = input$k, ncp = 0), 0, input$k + 10, ylim = c(0, 0.6),
            ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
      curve(pchisq(x, df  = input$k, ncp = 0),0, input$k + 10, ylim = c(0, 1), type = "l", main = "Fu
            ylab = "F(x)", lwd = 2, col = "gray")})

    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(pchisq(input$a, df  = input$k, ncp = 0), d

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pchisq(input$b, df  = input$k, ncp = 0

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((pchisq(input$b, df  = in
    })

    output$Px_a <- renderPlot({curve(pchisq(x, df  = input$k, ncp = 0),0, input$k + 10, ylim = c(0, 1
                                     ylab = "F(x)", lwd = 2, col = "gray")

      x <- seq(0,input$a, by=0.05)
```

```
    x1 <- c(x, seq(input$a,0, by=-0.05))
    y <- c(c(pchisq(x, df  = input$k, ncp = 0)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "pink")

    x <- seq(input$b, input$k+10, by=0.05)
    x1 <- c(x, seq(input$k+10, input$b, by=-0.05))
    y <- c(c(pchisq(x, df  = input$k, ncp = 0)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "#FF00FF")

    x<-seq(input$a, input$b, by=0.05)
    x1 <- c(x, seq(input$b, input$a, by=-0.05))
    y <- c(c(pchisq(x, df  = input$k, ncp = 0)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "purple")
  })
}
```

5. Repartitia Exponentiala

```
if(input$var == 5)
{shinyjs::hide(id = "uniforma")
  shinyjs::hide(id = "binomiala")
  shinyjs::hide(id = "cauchy")
  shinyjs::hide(id = "Chy-Squagray")
  shinyjs::show(id = "exponentiala")
  shinyjs::hide(id = "fisher")
  shinyjs::hide(id = "gamma")
  shinyjs::hide(id = "log-normala")
  shinyjs::hide(id = "hipergeometrica")
  shinyjs::hide(id = "logistica")
  shinyjs::hide(id = "poisson")
  shinyjs::hide(id = "student")
  shinyjs::hide(id = "normala")
  shinyjs::hide(id = "Beta")
  shinyjs::hide(id = "weibull")
  output$functii <- renderPlot({par(mfrow = c(1, 2))
    curve(dexp(x, rate  = input$lambda), 0, 5, ylim = c(0, 1.5),
          ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
    curve(pexp(x, rate  = input$lambda),0, 5, ylim = c(0, 1), type = "l", main = "Functia de repart
          ylab = "F(x)", lwd = 2, col = "gray")})

  output$Px_a_val <- renderText({
    paste("P( X <= a ) = P( X <=", input$a, ") =", round(pexp(input$a, rate  = input$lambda), digit

  })
  output$Px_b_val <- renderText({
    paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pexp(input$b, rate  = input$lambda), d

  })
  output$Px_a_b_val <- renderText({
    paste("P( a <= X <= b ) = P( ",input$a,"<= X <=", input$b, ") =", round((pexp(input$b, rate  =
  })
```

```
    output$Px_a <- renderPlot({curve(pexp(x, rate  = input$lambda),0, 5, ylim = c(0, 1), type = "l", m
                                     ylab = "F(x)", lwd = 2, col = "gray")

      x <- seq(0,input$a, by=0.05)
      x1 <- c(x, seq(input$a,0, by=-0.05))
      y <- c(c(pexp(x, rate  = input$lambda)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "pink")

      x <- seq(input$b, 5, by=0.05)
      x1 <- c(x, seq(5, input$b, by=-0.05))
      y <- c(c(pexp(x, rate  = input$lambda)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "#FF00FF")

      x<-seq(input$a, input$b, by=0.05)
      x1 <- c(x, seq(input$b, input$a, by=-0.05))
      y <- c(c(pexp(x, rate  = input$lambda)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

6. Repartitia Fisher

```
  if(input$var == 6)
  {shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::show(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      curve(df(x, df1 = input$d1, df2 = input$d2), 0, 5, ylim = c(0, 2.5),
            ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
      curve(pf(x, df1 = input$d1, df2 = input$d2),0, 5, ylim = c(0, 1), type = "l", main = "Functia d
            ylab = "F(x)", lwd = 2, col = "gray")})

    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(pf(input$a, df1 = input$d1, df2 = input$d2)

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pf(input$b, df1 = input$d1, df2 = inpu

    })
```

```
  output$Px_a_b_val <- renderText({
    paste("P( a <= X <= b ) = P( ",input$a,"<= X <=", input$b, ") =", round((pf(input$b, df1 = inpu
  })

  output$Px_a <- renderPlot({curve(pf(x, df1 = input$d1, df2 = input$d2),0, 5, ylim = c(0, 1), type
                                   ylab = "F(x)", lwd = 2, col = "gray")

    x <- seq(0,input$a, by=0.05)
    x1 <- c(x, seq(input$a,0, by=-0.05))
    y <- c(c(pf(x, df1 = input$d1, df2 = input$d2)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "pink")

    x <- seq(input$b, 5, by=0.05)
    x1 <- c(x, seq(5, input$b, by=-0.05))
    y <- c(c(pf(x, df1 = input$d1, df2 = input$d2)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "#FF00FF")

    x<-seq(input$a, input$b, by=0.05)
    x1 <- c(x, seq(input$b, input$a, by=-0.05))
    y <- c(c(pf(x, df1 = input$d1, df2 = input$d2)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "purple")
  })
}
```

7. Repartitia Gamma

```
if(input$var == 7)
{shinyjs::hide(id = "uniforma")
  shinyjs::hide(id = "binomiala")
  shinyjs::hide(id = "cauchy")
  shinyjs::hide(id = "Chy-Squagray")
  shinyjs::hide(id = "exponentiala")
  shinyjs::hide(id = "fisher")
  shinyjs::show(id = "gamma")
  shinyjs::hide(id = "log-normala")
  shinyjs::hide(id = "hipergeometrica")
  shinyjs::hide(id = "logistica")
  shinyjs::hide(id = "poisson")
  shinyjs::hide(id = "student")
  shinyjs::hide(id = "normala")
  shinyjs::hide(id = "Beta")
  shinyjs::hide(id = "weibull")
  output$functii <- renderPlot({par(mfrow = c(1, 2))
    curve(dgamma(x, shape = input$alpha, rate = input$beta, scale = 1/input$beta), 0, 20, ylim = c(
          ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
    curve(pgamma(x, shape = input$alpha, rate = input$beta, scale = 1/input$beta),0, 20, ylim = c(0
          ylab = "F(x)", lwd = 2, col = "gray")})

  output$Px_a_val <- renderText({
    paste("P( X <= a ) = P( X <=", input$a, ") =", round(pgamma(input$a, shape = input$alpha, rate =

  })
```

```
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pgamma(input$b, shape = input$alpha, ra

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P( ",input$a,"<= X <=", input$b, ") =", round((pgamma(input$b, shape =
    })

    output$Px_a <- renderPlot({curve(pgamma(x, shape = input$alpha, rate = input$beta, scale = 1/input
                                     ylab = "F(x)", lwd = 2, col = "gray")

      x <- seq(0,input$a, by=0.05)
      x1 <- c(x, seq(input$a,0, by=-0.05))
      y <- c(c(pgamma(x, shape = input$alpha, rate = input$beta, scale = 1/input$beta)),seq(0,0, leng
      polygon(x1, y, col = "pink")

      x <- seq(input$b, 20, by=0.05)
      x1 <- c(x, seq(20, input$b, by=-0.05))
      y <- c(c(pgamma(x, shape = input$alpha, rate = input$beta, scale = 1/input$beta)),seq(0,0, leng
      polygon(x1, y, col = "#FF00FF")

      x<-seq(input$a, input$b, by=0.05)
      x1 <- c(x, seq(input$b, input$a, by=-0.05))
      y <- c(c(pgamma(x, shape = input$alpha, rate = input$beta, scale = 1/input$beta)),seq(0,0, leng
      polygon(x1, y, col = "purple")
    })
  }
```

8. Repartitia Hipergeometrica

```
  if(input$var == 8)
  { shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "log-normala")
    shinyjs::show(id = "hipergeometrica")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      plot(seq(0,60,1),dhyper(seq(0,60,1), m = input$m_, n = input$n_, k = input$k_),type="b", xlab =
           lwd = 2, col = "gray")
      plot(seq(0,60,1),phyper(seq(0,60,1), m = input$m_, n = input$n_, k = input$k_),type="b", xlab =
           main = "Functia de repartitie", lwd = 2, col = "gray")})
    output$Px_a_val <- renderText({
```

```
    paste("P( X <= a ) = P( X <=", input$a, ") =", round(phyper(input$a, m = input$m_, n = input$n_
  })
  output$Px_b_val <- renderText({
    paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(phyper(input$b, m = input$m_, n = inpu
  })
  output$Px_a_b_val <- renderText({
    paste("P( a <= X <= b ) = P( ",input$a,"<= X <=", input$b, ") =", round((phyper(input$b, m = inp
  })
  output$Px_a <- renderPlot({plot(seq(0,60,1),phyper(seq(0,60,1), n = input$m_, m = input$n_, k = in
                                  lwd = 2, col = "gray")

    x <- seq(0,input$a, by=0.001)
    x1 <- c(x, seq(input$a,0, by=-0.001))
    y <- c(c(phyper(x, m = input$m_, n = input$n_, k = input$k_)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "pink")

    x <- seq(input$b, 60, by=0.001)
    x1 <- c(x, seq(60, input$b, by=-0.001))
    y <- c(c(phyper(x, m = input$m_, n = input$n_, k = input$k_)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "#FF00FF")

    x<-seq(input$a, input$b, by=0.001)
    x1 <- c(x, seq(input$b, input$a, by=-0.001))
    y <- c(c(phyper(x, m = input$m_, n = input$n_, k = input$k_)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "purple")
  })
}
```

9. Repartitia Log-Normala

```
if(input$var == 9)
{ shinyjs::hide(id = "uniforma")
  shinyjs::hide(id = "binomiala")
  shinyjs::hide(id = "cauchy")
  shinyjs::hide(id = "Chy-Squagray")
  shinyjs::hide(id = "exponentiala")
  shinyjs::hide(id = "fisher")
  shinyjs::hide(id = "gamma")
  shinyjs::hide(id = "hipergeometrica")
  shinyjs::show(id = "log-normala")
  shinyjs::hide(id = "logistica")
  shinyjs::hide(id = "poisson")
  shinyjs::hide(id = "student")
  shinyjs::hide(id = "normala")
  shinyjs::hide(id = "Beta")
  shinyjs::hide(id = "weibull")
  output$functii <- renderPlot({par(mfrow = c(1, 2))
    curve(dlnorm(x, meanlog = input$mu, sdlog = input$teta), 0, 2.5, ylim = c(0, 2),
          ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
    curve(plnorm(x, meanlog = input$mu, sdlog = input$teta),0, 2.5, ylim = c(0, 1.5), type = "l", ma
```

```
            ylab = "F(x)", lwd = 2, col = "gray")})
  output$Px_a_val <- renderText({
    paste("P( X <= a ) = P( X <=", input$a, ") =", round(plnorm(input$a, meanlog = input$mu, sdlog =

  })
  output$Px_b_val <- renderText({
    paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(plnorm(input$b,meanlog = input$mu, sdl

  })
  output$Px_a_b_val <- renderText({
    paste("P( a <= X <= b ) = P( ",input$a,"<= X <=", input$b, ") =", round((plnorm(input$b, meanlo
  })
  output$Px_a <- renderPlot({curve(plnorm(x, meanlog = input$mu, sdlog = input$teta),0, 2.5, ylim =
                                   ylab = "F(x)", lwd = 2, col = "gray")

    x <- seq(0,input$a, by=0.001)
    x1 <- c(x, seq(input$a,0, by=-0.001))
    y <- c(c(plnorm(x, meanlog = input$mu, sdlog = input$teta)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "pink")

    x <- seq(input$b, 2.5, by=0.001)
    x1 <- c(x, seq(2.5, input$b, by=-0.001))
    y <- c(c(plnorm(x, meanlog = input$mu, sdlog = input$teta)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "#FF00FF")

    x<-seq(input$a, input$b, by=0.001)
    x1 <- c(x, seq(input$b, input$a, by=-0.001))
    y <- c(c(plnorm(x, meanlog = input$mu, sdlog = input$teta)),seq(0,0, length = length(x)))
    polygon(x1, y, col = "purple")
  })
}
```

10. Repartitia Logistica

```
if(input$var == 10)
{ shinyjs::hide(id = "uniforma")
  shinyjs::hide(id = "binomiala")
  shinyjs::hide(id = "cauchy")
  shinyjs::hide(id = "Chy-Squagray")
  shinyjs::hide(id = "exponentiala")
  shinyjs::hide(id = "fisher")
  shinyjs::hide(id = "gamma")
  shinyjs::hide(id = "hipergeometrica")
  shinyjs::hide(id = "log-normala")
  shinyjs::show(id = "logistica")
  shinyjs::hide(id = "poisson")
  shinyjs::hide(id = "student")
  shinyjs::hide(id = "normala")
  shinyjs::hide(id = "Beta")
  shinyjs::hide(id = "weibull")
  output$functii <- renderPlot({par(mfrow = c(1, 2))
    curve(dlogis(x, location  = input$tt, scale = input$ss), -5, 20, ylim = c(0, 0.3),
```

```
            ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
      curve(plogis(x, location  = input$tt, scale = input$ss),-5 ,20, ylim = c(0, 1), type = "l", mai
            ylab = "F(x)", lwd = 2, col = "gray")})

    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(plogis(input$a, location  = input$tt, scal

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(plogis(input$b, location  = input$tt,

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((plogis(input$b, locatio
    })

    output$Px_a <- renderPlot({curve(plogis(x, location  = input$tt, scale = input$ss),-5, 20, ylim =
                                     ylab = "F(x)", lwd = 2, col = "gray")

      x <- seq(-5,input$a, by=0.2)
      x1 <- c(x, seq(input$a,-5, by=-0.2))
      y <- c(c(plogis(x, location  = input$tt, scale = input$ss)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "pink")

      x <- seq(input$b, 20, by=0.2)
      x1 <- c(x, seq(20, input$b, by=-0.2))
      y <- c(c(plogis(x, location  = input$tt, scale = input$ss)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "#FF00FF")

      x<-seq(input$a, input$b, by=0.2)
      x1 <- c(x, seq(input$b, input$a, by=-0.2))
      y <- c(c(plogis(x, location  = input$tt, scale = input$ss)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

11. Repartitia Normala

```
  if(input$var == 11)
  { shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::show(id = "normala")
```

```
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      curve(dnorm(x, mean = input$mu_, sd = sqrt(input$teta_)), -5, 5, ylim = c(0, 1),
            ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
      curve(pnorm(x, mean = input$mu_, sd = sqrt(input$teta_)),-5, 5, ylim = c(0, 1), type = "l", mai
            ylab = "F(x)", lwd = 2, col = "gray")})
    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(pnorm(input$a, mean = input$mu_, sd = sqrt

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pnorm(input$b,mean = input$mu_, sd = s

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P( ",input$a,"<= X <=", input$b, ") =", round((pnorm(input$b, mean =
    })
    output$Px_a <- renderPlot({curve(pnorm(x, mean = input$mu_, sd = sqrt(input$teta_)),-5, 5, ylim =
                                     ylab = "F(x)", lwd = 2, col = "gray")

      x <- seq(-5,input$a, by=0.001)
      x1 <- c(x, seq(input$a,-5, by=-0.001))
      y <- c(c(pnorm(x, mean = input$mu_, sd = sqrt(input$teta_))),seq(0,0, length = length(x)))
      polygon(x1, y, col = "pink")

      x <- seq(input$b, 5, by=0.001)
      x1 <- c(x, seq(5, input$b, by=-0.001))
      y <- c(c(pnorm(x, mean = input$mu_, sd = sqrt(input$teta_))),seq(0,0, length = length(x)))
      polygon(x1, y, col = "#FF00FF")

      x<-seq(input$a, input$b, by=0.001)
      x1 <- c(x, seq(input$b, input$a, by=-0.001))
      y <- c(c(pnorm(x, mean = input$mu_, sd = sqrt(input$teta_))),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

12. Repartitia Poisson

```
  { shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "logistica")
    shinyjs::show(id = "poisson")
    shinyjs::hide(id = "student")
```

```
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      plot(seq(0,20,1),dpois(seq(0,20,1), lambda = input$lambda_),type="b", xlab = "x", ylab = "f(x)"
           lwd = 2, col = "gray")
      plot(seq(0,20,1),ppois(seq(0,20,1), lambda = input$lambda_),type="p", xlab = "x", ylab = "F(x)"
           lwd = 2, col = "gray")})
    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(ppois(input$a,lambda = input$lambda_), dig

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(ppois(input$b, lambda = input$lambda_)

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P( ",input$a,"<= X <=", input$b, ") =", round((ppois(input$b,lambda =
    })
    output$Px_a <- renderPlot({plot(seq(0,20,1),ppois(seq(0,20,1), lambda = input$lambda_),type="p", 
                                    lwd = 2, col = "gray")

      x <- seq(0,input$a, by=0.001)
      x1 <- c(x, seq(input$a,0, by=-0.001))
      y <- c(c(ppois(x, lambda = input$lambda_)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "pink")

      x <- seq(input$b, 20, by=0.001)
      x1 <- c(x, seq(20, input$b, by=-0.001))
      y <- c(c(ppois(x,lambda = input$lambda_)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "#FF00FF")

      x<-seq(input$a, input$b, by=0.001)
      x1 <- c(x, seq(input$b, input$a, by=-0.001))
      y <- c(c(ppois(x, lambda = input$lambda_)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

13. Repartitia Student

```
  if(input$var == 13)
  { shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "logistica")
```

```
    shinyjs::hide(id = "poisson")
    shinyjs::show(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      curve(dt(x, df = input$d), -4, 4, ylim = c(0, 0.4),
            ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
      curve(pt(x, df = input$d),-4 ,4, ylim = c(0, 1), type = "l", main = "Functia de repartitie",
            ylab = "F(x)", lwd = 2, col = "gray")})

    output$Px_a_val <- renderText({
      paste("P( X <= a ) = P( X <=", input$a, ") =", round(pt(input$a, df = input$d), digits = 3))

    })
    output$Px_b_val <- renderText({
      paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pt(input$b, df = input$d), digits = 3)

    })
    output$Px_a_b_val <- renderText({
      paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((pt(input$b, df = input$
    })

    output$Px_a <- renderPlot({curve(pt(x, df = input$d),-4, 4, ylim = c(0, 1), type = "l", main = "P
                                     ylab = "F(x)", lwd = 2, col = "gray")

      x <- seq(-4,input$a, by=0.1)
      x1 <- c(x, seq(input$a,-4, by=-0.1))
      y <- c(c(pt(x, df = input$d)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "pink")

      x <- seq(input$b, 4, by=0.1)
      x1 <- c(x, seq(4, input$b, by=-0.1))
      y <- c(c(pt(x, df = input$d)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "#FF00FF")

      x<-seq(input$a, input$b, by=0.1)
      x1 <- c(x, seq(input$b, input$a, by=-0.1))
      y <- c(c(pt(x, df = input$d)),seq(0,0, length = length(x)))
      polygon(x1, y, col = "purple")
    })
  }
```

14. Repartitia Uniforma

```
  if(input$var == 14)
  {shinyjs::show(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
```

```
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
    shinyjs::hide(id = "weibull")
    output$functii <- renderPlot({par(mfrow = c(1, 2))
      curve(dunif(x, min = input$minU, max = input$maxU), input$minU-2, input$maxU+2, ylim = c(0, 1),
            ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
      curve(punif(x, min = input$minU, max = input$maxU),input$minU-2, input$maxU+2, ylim = c(0, 1),
            ylab = "F(x)", lwd = 2, col = "gray")
      output$Px_a_val <- renderText({
        paste("P( X <= a) = P( X <=", input$a, ") =", round(punif(input$a, min = input$minU, max = inp

      })
      output$Px_b_val <- renderText({
        paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(punif(input$b, min = input$minU, max

      })
      output$Px_a_b_val <- renderText({
        paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((punif(input$b, min = i
      })
      output$Px_a <- renderPlot({curve(punif(x, min = input$minU, max = input$maxU),input$minU-2, inpu
                                       ylab = "F(x)", lwd = 2, col = "gray")

        polygon(c(input$minU,input$a,input$a), c(0, 0, punif(input$a, min = input$minU, max = input$ma
        polygon(c(input$b,input$maxU,input$maxU,input$b), c(0, 0,1, punif(input$b, min = input$minU, m
        polygon(c(input$a,input$b,input$b,input$a), c(0, 0,punif(input$b, min = input$minU, max = inpu
      })
    })
  }
```

15. Repartitia Weibull

```
  if(input$var == 15)
  { shinyjs::hide(id = "uniforma")
    shinyjs::hide(id = "binomiala")
    shinyjs::hide(id = "cauchy")
    shinyjs::hide(id = "Chy-Squagray")
    shinyjs::hide(id = "exponentiala")
    shinyjs::hide(id = "fisher")
    shinyjs::hide(id = "gamma")
    shinyjs::hide(id = "hipergeometrica")
    shinyjs::hide(id = "log-normala")
    shinyjs::hide(id = "logistica")
    shinyjs::hide(id = "poisson")
    shinyjs::hide(id = "student")
    shinyjs::hide(id = "normala")
    shinyjs::hide(id = "Beta")
```

```
      shinyjs::show(id = "weibull")
      output$functii <- renderPlot({par(mfrow = c(1, 2))
        curve(dweibull(x, shape = input$kk, scale = 1), 0, 2.5, ylim = c(0, 2.5),
              ylab = "f(x)",col = "gray", lwd = 2, main = "Functia de densitate")
        curve(pweibull(x, shape = input$kk, scale = 1),0 ,2.5, ylim = c(0, 1), type = "l", main = "Func
              ylab = "F(x)", lwd = 2, col = "gray")})

      output$Px_a_val <- renderText({
        paste("P( X <= a ) = P( X <=", input$a, ") =", round(pweibull(input$a, shape = input$kk, scale =

      })
      output$Px_b_val <- renderText({
        paste("P( X >= b ) = P( X >=", input$b, ") =", 1 - round(pweibull(input$b, shape = input$kk, sc

      })
      output$Px_a_b_val <- renderText({
        paste("P( a <= X <= b ) = P(",input$a,"<= X <=", input$b, ") =", round((pweibull(input$b, shape
      })

      output$Px_a <- renderPlot({curve(pweibull(x, shape = input$kk, scale = 1),0, 2.5, ylim = c(0, 1),
                                       ylab = "F(x)", lwd = 2, col = "gray")

        x <- seq(0,input$a, by=0.01)
        x1 <- c(x, seq(input$a,0, by=-0.01))
        y <- c(c(pweibull(x, shape = input$kk, scale = 1)),seq(0,0, length = length(x)))
        polygon(x1, y, col = "pink")

        x <- seq(input$b, 2.5, by=0.01)
        x1 <- c(x, seq(2.5, input$b, by=-0.01))
        y <- c(c(pweibull(x,shape = input$kk, scale = 1)),seq(0,0, length = length(x)))
        polygon(x1, y, col = "#FF00FF")

        x<-seq(input$a, input$b, by=0.01)
        x1 <- c(x, seq(input$b, input$a, by=-0.01))
        y <- c(c(pweibull(x, shape = input$kk, scale = 1)),seq(0,0, length = length(x)))
        polygon(x1, y, col = "purple")
      })
    }
  })
}
```