



**CONCORSO PUBBLICO PER TITOLI ED ESAMI PER LA COPERTURA A TEMPO DETERMINATO  
DI N. 1 UNITA' DI PERSONALE NEL PROFILO PROFESSIONALE DI COLLABORATORE  
PROFESSIONALE DELLA RICERCA SANITARIA, CAT. D CON LAUREA IN STATISTICA**

**PROVA ORALE N° 1**

1. Il candidato definisca i concetti di matrice e array in R
2. Il candidato commenti il seguente codice:

```
library(shiny)
ui <- fluidPage(
  titlePanel("Reactivity"),
  sidebarLayout(
    sidebarPanel(
      textInput(inputId = "caption", label = "Caption:", value = "Data Summary"),
      selectInput(inputId = "dataset", label = "Choose a dataset:", choices = c("rock",
"pressure", "cars")),
      numericInput(inputId = "obs", label = "Number of observations to view:", value = 10)),
    mainPanel(
      h3(textOutput("caption", container = span)), verbatimTextOutput("summary"),
      tableOutput("view"))))

server <- function(input, output) {
datasetInput <- reactive({switch(input$dataset, "rock" = rock, "pressure" = pressure, "cars"
= cars)})

output$caption <- renderText({ input$caption})

output$summary <- renderPrint({

dataset <- datasetInput()

summary(dataset)
})

output$view <- renderTable({
head(datasetInput(), n = input$obs)})
}

shinyApp(ui, server)
```

**PROVA ORALE N° 2**

3. Il candidato descriva sinteticamente le differenze tra approccio Frequentista e Bayesiano.
4. Il candidato commenti il seguente codice:

```
Library(shiny)

vars <- setdiff(names(iris), "Species")

UI
pageWithSidebar(
  headerPanel('Iris k-means clustering'),
  sidebarPanel(
    selectInput('xcol', 'X Variable', vars),
```



```
selectInput('ycol', 'Y Variable', vars, selected = vars[[2]]),  
numericInput('clusters', 'Cluster count', 3, min = 1, max = 9)  
)  
mainPanel(  
  plotOutput('plot1')  
)  
)
```

#### SERVER

```
function(input, output, session) {  
  selectedData <- reactive({  
    iris[, c(input$xcol, input$ycol)]  
  })  
  
  clusters <- reactive({  
    kmeans(selectedData(), input$clusters)  
  })  
  
  output$plot1 <- renderPlot({  
    palette(c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3",  
             "#FF7F00", "#FFFF33", "#A65628", "#F781BF", "#999999"))  
  
    par(mar = c(5.1, 4.1, 0, 1))  
    plot(selectedData(),  
          col = clusters()$cluster,  
          pch = 20, cex = 3)  
    points(clusters()$centers, pch = 4, cex = 4, lwd = 4)  
  })  
}
```

### PROVA ORALE N° 3

1. Il candidato spieghi Cos'è una variabile casuale e indicare le diverse tipologie.
2. Il candidato commenti il seguente codice:

#### UI

```
bootstrapPage(  
  
  selectInput(inputId = "n_breaks",  
              label = "Number of bins in histogram (approximate):",  
              choices = c(10, 20, 35, 50),  
              selected = 20),  
  checkboxInput(inputId = "individual_obs",  
                label = strong("Show individual observations"),  
                value = FALSE),  
  
  checkboxInput(inputId = "density",  
                label = strong("Show density estimate"),  
                value = FALSE),  
  plotOutput(outputId = "main_plot", height = "300px"),  
  conditionalPanel(condition = "input.density == true",  
                    sliderInput(inputId = "bw_adjust",  
                                label = "Bandwidth adjustment:",  
                                min = 0.2, max = 2, value = 1, step = 0.2)  
  )  
)
```

#### SERVER

```
function(input, output) {
```



```
output$main_plot <- renderPlot({  
  
  hist(faithful$eruptions,  
       probability = TRUE,  
       breaks = as.numeric(input$n_breaks),  
       xlab = "Duration (minutes)",  
       main = "Geyser eruption duration")  
  
  if (input$individual_obs) {  
    rug(faithful$eruptions)  
  }  
  if (input$density) {  
    dens <- density(faithful$eruptions,  
                    adjust = input$bw_adjust)  
    lines(dens, col = "blue")  
  }  
  
})  
}
```

#### PROVA ORALE N° 4

1. Il candidato spieghi quando si utilizza la regressione lineare semplice? Quali sono gli assunti?
2. Il candidato commenti il seguente codice:

```
library(shiny)  
ui <- fluidPage(  
  titlePanel("Tabsets"),  
  sidebarLayout(  
    sidebarPanel(  
      radioButtons("dist", "Distribution type:",  
                  c("Normal" = "norm",  
                    "Uniform" = "unif",  
                    "Log-normal" = "lnorm",  
                    "Exponential" = "exp")),  
      br(),  
      sliderInput("n",  
                 "Number of observations:",  
                 value = 500,  
                 min = 1,  
                 max = 1000)  
    ),  
    mainPanel(  
      tabsetPanel(type = "tabs",  
                  tabPanel("Plot", plotOutput("plot")),  
                  tabPanel("Summary", verbatimTextOutput("summary")),  
                  tabPanel("Table", tableOutput("table"))  
                )))  
  )))  
server <- function(input, output) {  
  d <- reactive({  
    dist <- switch(input$dist,  
                  norm = rnorm,  
                  unif = runif,  
                  lnorm = rlnorm,  
                  exp = rexp,  
                  rnorm)  
  
    dist(input$n)  
  })  
  output$plot <- renderPlot({  
    dist <- input$dist
```



**ISTITUTO ZOOPROFILATTICO SPERIMENTALE  
DELLA LOMBARDIA E DELL'EMILIA ROMAGNA  
"BRUNO UBERTINI"**  
(ENTE SANITARIO DI DIRITTO PUBBLICO)

---

```
n <- input$n

hist(d(),
     main = paste("r", dist, "(", n, ")", sep = ""),
     col = "#75AADB", border = "white")
})
output$summary <- renderPrint({
  summary(d())
})
output$table <- renderTable({
  d()
})
}
shinyApp(ui, server)
```