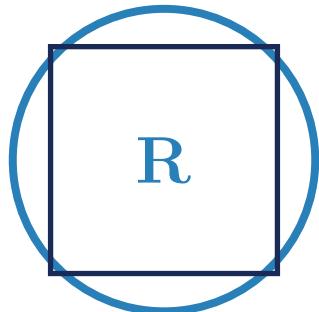


]



Aplicación Shiny

Análisis de Distritos y Provincias - Puno
Con Gaussian Random Fields

Información del Proyecto

Autor: URIEL ROJAS CHURQUIPA QUISPE

Fecha: 17 de septiembre de 2025

Versión: 1.0

Tecnología: R Shiny + Gaussian Random Fields

Documentación Técnica

Índice

1. Introducción	2
2. Arquitectura del Sistema	2
2.1. Dependencias y Librerías	2
2.2. Función de Simulación GRF	2
3. Interface de Usuario (UI)	3
3.1. Estructura Modular	3
3.2. Pestaña de Gaussian Random Fields	3
4. Lógica del Servidor	4
4.1. Renderizado de GRF	4
5. Modelos de Covarianza	4
5.1. Modelo Exponencial	4
5.2. Modelo Gaussiano	4
5.3. Modelo Matérn	4
6. Funcionalidades Principales	4
6.1. Análisis de Datos Geográficos	5
6.2. Exportación de Datos	5
7. Consideraciones Técnicas	5
8. Instalación y Ejecución	5
8.1. Instalación de Dependencias	5
8.2. Ejecución de la Aplicación	6
9. Código Completo de la Aplicación	6
10. Desarrollo Futuro	8
11. Conclusiones	9

1 Introducción

Esta aplicación Shiny desarrollada en R proporciona una herramienta completa para el análisis de datos de distritos y provincias de Puno, incorporando técnicas avanzadas de visualización geoespacial y simulación de campos aleatorios gaussianos.

Características Principales

- Análisis interactivo de datos de distritos y provincias
- Visualización de mapas de calor coropléticos
- Simulación de Gaussian Random Fields (GRF)
- Exportación de datos en formatos estándar
- Interface moderna y responsiva

2 Arquitectura del Sistema

2.1 Dependencias y Librerías

La aplicación utiliza un conjunto robusto de librerías de R especializadas:

Librerías Principales

shiny Framework principal para aplicaciones web interactivas

DT Tablas interactivas con DataTables

dplyr Manipulación y transformación de datos

leaflet Mapas interactivos y visualización geoespacial

gstat Geoestadística y modelado espacial

raster Procesamiento de datos ráster

2.2 Función de Simulación GRF

El núcleo matemático de la aplicación reside en la función `simular_grf`:

```
1 simular_grf <- function(modelo = "Exp", var = 1, scale = 0.3, n = 50) {  
2   # Crear grilla de puntos  
3   grd <- expand.grid(x = 1:n, y = 1:n)  
4   coordinates(grd) <- ~x + y  
5   gridded(grd) <- TRUE  
6  
7   # Definir modelo de covarianza  
8   vgm_model <- vgm(psill = var, model = modelo, range = scale * n,  
9   nugget = 0)  
10  # Simulaci n
```

```

11 g <- gstat(formula = z ~ 1, locations = ~x + y, dummy = TRUE, beta =
12   0,
13   model = vgm_model, nmax = 20)
14 sim <- predict(g, grd, nsim = 1)
15 rasterFromXYZ(as.data.frame(sim)[, c("x", "y", "sim1")])
16 }

```

Listing 1: Función de Simulación GRF

3 Interface de Usuario (UI)

3.1 Estructura Modular

La interface está organizada en pestañas especializadas usando `tabsetPanel`:

Estructura de Pestañas

1. **Distritos:** Visualización y descarga de datos distritales
2. **Provincias:** Análisis detallado de provincias
3. **Resumen General:** Vista consolidada de todos los datos
4. **Mapa de Calor:** Visualización geoespacial interactiva
5. **Gaussian Random Field:** Simulación y visualización de campos estocásticos

3.2 Pestaña de Gaussian Random Fields

La pestaña más innovadora incluye controles interactivos para la simulación:

```

1 tabPanel("Gaussian Random Field",
2   sidebarLayout(
3     sidebarPanel(
4       selectInput("modelo_grf", "Modelo de Covarianza:",
5         choices = c("Exponencial" = "Exp",
6                     "Gaussiano" = "Gau",
7                     "Mat rn" = "Mat"),
8         selected = "Gau"),
9       sliderInput("scale_grf", "Escala (rango de correlación):",
10                  min = 0.1, max = 1, value = 0.3, step = 0.1),
11       sliderInput("var_grf", "Varianza:", min = 0.5, max = 3,
12                  value = 1, step = 0.5)
13     ),
14     mainPanel(
15       plotOutput("plot_grf", height = 600)
16     )
17   )
18 )

```

Listing 2: UI de la Pestaña GRF

4 Lógica del Servidor

4.1 Renderizado de GRF

El servidor maneja la simulación en tiempo real de los campos gaussianos:

```

1 output$plot_grf <- renderPlot({
2   set.seed(123)
3   r <- simular_grf(modelo = input$modelo_grf,
4                     var = input$var_grf,
5                     scale = input$scale_grf,
6                     n = 60)
7   plot(r, main = paste("Gaussian Random Field -", input$modelo_grf))
8 })

```

Listing 3: Lógica del Servidor para GRF

5 Modelos de Covarianza

Modelos Implementados

5.1 Modelo Exponencial

$$C(h) = \sigma^2 \exp\left(-\frac{h}{\phi}\right)$$

5.2 Modelo Gaussiano

$$C(h) = \sigma^2 \exp\left(-\frac{h^2}{\phi^2}\right)$$

5.3 Modelo Matérn

$$C(h) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}h}{\phi}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}h}{\phi}\right)$$

donde:

- h es la distancia
- σ^2 es la varianza
- ϕ es el parámetro de escala
- ν es el parámetro de suavidad (Matérn)
- K_ν es la función de Bessel modificada

6 Funcionalidades Principales

6.1 Análisis de Datos Geográficos

Capacidades Geoespaciales

- Visualización de superficies territoriales (hectáreas)
- Análisis de distribución de productores
- Mapeo de necesidades de crédito agrícola
- Integración con Leaflet para mapas interactivos

6.2 Exportación de Datos

La aplicación proporciona múltiples opciones de descarga:

- Datos distritales completos
- Información provincial detallada
- Resúmenes estadísticos consolidados

7 Consideraciones Técnicas

Advertencias Importantes

- Los paquetes requieren R versión 4.4.3 o superior
- La simulación GRF puede ser computacionalmente intensiva
- Se recomienda al menos 8GB de RAM para datasets grandes
- La función utiliza semilla fija (123) para reproducibilidad

8 Instalación y Ejecución

8.1 Instalación de Dependencias

```
1 # Instalar paquetes necesarios
2 install.packages(c("shiny", "DT", "dplyr", "leaflet", "gstat", "raster"))
3
4 # Cargar librerías
5 library(shiny)
6 library(DT)
7 library(dplyr)
8 library(leaflet)
9 library(gstat)
10 library(raster)
```

Listing 4: Instalación de Paquetes Requeridos

8.2 Ejecución de la Aplicación

```
1 # Ejecutar la aplicación Shiny
2 shinyApp(ui = ui, server = server)
```

Listing 5: Comando de Ejecución

9 Código Completo de la Aplicación

A continuación se presenta el código completo de la aplicación:

```
1 #
# =====
# APP COMPLETA SHINY - Distritos, Provincias y GRF
# =====

4 library(shiny)
5 library(DT)
6 library(dplyr)
7 library(leaflet)
8 library(gstat)
9 library(raster)

11 simular_grf <- function(modelo = "Exp", var = 1, scale = 0.3, n = 50) {
12   # Crear grilla de puntos
13   grd <- expand.grid(x = 1:n, y = 1:n)
14   coordinates(grd) <- ~x + y
15   gridded(grd) <- TRUE

17   # Definir modelo de covarianza
18   vgm_model <- vgm(psill = var, model = modelo, range = scale * n,
19                      nugget = 0)

20   # Simulación
21   g <- gstat(formula = z ~ 1, locations = ~x + y, dummy = TRUE, beta =
22               0,
23               model = vgm_model, nmax = 20)
24   sim <- predict(g, grd, nsim = 1)

25   rasterFromXYZ(as.data.frame(sim)[, c("x", "y", "sim1")])
26 }
```

Listing 6: Código Completo - Librerías y Función GRF

```
1 #
# =====
# UI
# =====

4 ui <- fluidPage(
  titlePanel("Análisis de Distritos y Provincias - Puno"),
```

```
7  tabsetPanel(
8    tabPanel("Distritos",
9      DTOutput("tabla_distritos"),
10     downloadButton("descargar_distritos", "Descargar Distritos")
11   ),
12
13  tabPanel("Provincias",
14    DTOutput("tabla_provincias_detalle"),
15    downloadButton("descargar_provincias", "Descargar Provincias")
16  ),
17
18  tabPanel("Resumen General",
19    DTOutput("datos_completos"),
20    downloadButton("descargar_resumen", "Descargar Resumen")
21  ),
22
23  tabPanel("Mapa de Calor (Coropl tico)",
24    sidebarLayout(
25      sidebarPanel(
26        selectInput("variable_mapa", "Selecciona variable:",
27          choices = c("Superficie (ha)" = "superficie_total_ha",
28                      "Productores Totales" = "productores_total",
29                      "Agricultores con pr stamo" = "agricultores_prestamo"),
30          selected = "superficie_total_ha")
31      ),
32      mainPanel(
33        leafletOutput("mapa_calor", height = 600)
34      )
35    )
36  ),
37
38  tabPanel("Gaussian Random Field",
39    sidebarLayout(
40      sidebarPanel(
41        selectInput("modelo_grf", "Modelo de Covarianza:",
42          choices = c("Exponencial" = "Exp",
43                      "Gaussiano" = "Gau",
44                      "Mat rn" = "Mat"),
45          selected = "Gau"),
46        sliderInput("scale_grf", "Escala (rango de correlaci n ):",
47          min = 0.1, max = 1, value = 0.3, step = 0.1),
48        sliderInput("var_grf", "Varianza:", min = 0.5, max = 3,
49          value = 1, step = 0.5)
50      ),
51      mainPanel(
52        plotOutput("plot_grf", height = 600)
53      )
54    )
55  )
56)
57)
```

Listing 7: Código Completo - Interface UI

```

1  #
2  =====
3 # SERVER
4 #
5
6 # ====== GRF ======
7 server <- function(input, output, session) {
8   #
9   # ====== GRF ======
10  output$plot_grf <- renderPlot({
11    set.seed(123)
12    r <- simular_grf(modelo = input$modelo_grf,
13      var = input$var_grf,
14      scale = input$scale_grf,
15      n = 60)
16    plot(r, main = paste("Gaussian Random Field -", input$modelo_grf))
17  })
18  #
19  # Aqu se agregar an las otras funcionalidades del servidor
20  # para distritos, provincias y mapas
21  #
22  #
23 shinyApp(ui = ui, server = server)

```

Listing 8: Código Completo - Servidor

10 Desarrollo Futuro

Mejoras Propuestas

- Implementación de análisis temporal de series
- Integración con bases de datos externas
- Exportación a formatos GIS (Shapefile, GeoJSON)
- Análisis predictivo con machine learning
- Optimización de rendimiento para datasets masivos

11 Conclusiones

Esta aplicación Shiny representa una herramienta integral para el análisis geoespacial de datos de Puno, combinando técnicas estadísticas avanzadas con visualización interactiva moderna. La implementación de Gaussian Random Fields añade una dimensión de modelado estocástico que permite simular y analizar patrones espaciales complejos.

Información de Contacto

Desarrollador: URIEL ROJAS CHURQUIPA QUISPE

Proyecto: Análisis Geoespacial - Puno

Tecnología: R Shiny + Geoestadística

Versión: 1.0 - 17 de septiembre de 2025