

Posicionamiento de Información:ANÁLISIS ESPACIAL DE PRODUCCIÓN AGRÍCOLA DEPARTAMENTO DE PUNO

Wladimir Aldo Carlosviza Amanqui

21 de octubre de 2025

Variables analizadas: Producción de Maíz Amarillo Duro, Quinua y Alfalfa

Fuente de datos: Encuesta Nacional Agropecuaria (ENA)

Código de departamento: CCDD = 21 (Puno)

Este documento presenta un análisis espacial de la producción agrícola en el departamento de Puno, utilizando datos de la Encuesta Nacional Agropecuaria. Se analizan tres productos clave: Maíz Amarillo Duro, Quinua y Alfalfa, cuantificando la producción en toneladas por distrito mediante visualizaciones interactivas georeferenciadas.

1. Librerías necesarias

```
library(tidyr)
library( )
library(htmlwidgets)
```

Interpretación: Se cargan tres librerías fundamentales: `tidyr` permite la reorganización de datos en formato ancho/largo, posibilita la creación de mapas interactivos basados en JavaScript, y `htmlwidgets` facilita el guardado de visualizaciones web autónomas.

2. Carga de datos

```
# ETAPA 1: CARGA DE DATOS
ena <- read.csv2("D:/10mo/estadistica espacial/03_CAP200AB.csv",
                stringsAsFactors = FALSE)
```

Interpretación: Se importa la base de datos ENA en formato CSV utilizando `read.csv2`, apropiado para archivos con punto y coma como separador. El argumento `stringsAsFactors = FALSE` evita la conversión automática de cadenas de texto a factores, preservando la flexibilidad en el manejo de datos categóricos.

3. Filtrado geográfico

```
# ETAPA 2: FILTRAR SOLO PUNO
ena_puno <- ena %>% filter(CCDD == 21)
```

Interpretación: Se aplica un filtro espacial para extraer únicamente los registros correspondientes al departamento de Puno, identificado con el código CCDD == 21. Esta segmentación permite concentrar el análisis en una región específica de interés agronómico.

4. Selección de productos agrícolas

```
# ETAPA 3: SELECCIONAR 5 PRODUCTOS AGRICOLAS
productos <- c("MAIZ AMARILLO DURO", "QUINUA", "ALFALFA")
ena_filtrado <- ena_puno %>% filter(P204_NOM %in% productos)
```

Interpretación: Se definen tres productos agrícolas estratégicos para el análisis: Maíz Amarillo Duro (cultivo comercial), Quinua (producto andino de exportación) y Alfalfa (forraje para ganadería). El filtrado mediante el operador `%in%` selecciona todos los registros que correspondan a alguno de estos productos.

5. Generación de centroides distritales

```
# ETAPA 4: GENERAR CENTROIDES SIMULADOS (LAT/LONG) POR DISTRITO
set.seed(123)
distritos <- ena_filtrado %>%
  distinct(NOMBREPV, NOMBREDI) %>%
  mutate(
    LATITUD = -15.5 + runif(n(), -0.5, 0.5),
    LONGITUD = -70 + runif(n(), -0.5, 0.5)
  )
```

Interpretación: Ante la ausencia de un shapefile oficial con geometrías distritales, se generan coordenadas de centroide simuladas para cada distrito único en la base de datos. Las coordenadas se distribuyen aleatoriamente alrededor de la ubicación aproximada de Puno ($-15,5$ latitud, -70 longitud) con variación de $\pm 0,5$ grados. La función `set.seed(123)` garantiza la reproducibilidad de los resultados al fijar la semilla del generador de números pseudoaleatorios.

6. Agregación de producción por distrito

```
# ETAPA 5: CALCULAR PRODUCCION POR DISTRITO Y PRODUCTO
ena_agregado <- ena_filtrado %>%
  group_by(NOMBREPV, NOMBREDI, P204_NOM) %>%
  summarise(produccion_ton = sum(P219_EQUIV_KG, na.rm = TRUE)/1000,
    .groups = "drop") %>%
  left_join(distritos, by = c("NOMBREPV", "NOMBREDI"))
```

Interpretación: Se calcula la producción total agregada por distrito y producto mediante la agrupación con `group_by` y la sumariación de la variable `P219_EQUIV_KG` (producción en kilogramos). Los valores se convierten a toneladas dividiendo entre 1000. Posteriormente, mediante `left_join`, se incorporan las coordenadas geográficas simuladas, generando así un dataset completo con información productiva y espacial.

7. Reestructuración de datos

```
# ETAPA 6: PIVOT WIDER (MANTENER LAT/LONG)
ena_wide <- ena_agregado %>%
  pivot_wider(
    id_cols = c(NOMBREP, NOMBREDI, LATITUD, LONGITUD),
    names_from = P204_NOM,
    values_from = produccion_ton
  )
```

Interpretación: Se aplica una transformación de formato largo a formato ancho mediante `pivot_wider`. En el dataset resultante, cada fila representa un distrito único con sus coordenadas, y cada producto agrícola constituye una columna independiente con su respectiva producción en toneladas. Esta estructura facilita el análisis multivariado y la comparación entre productos dentro de un mismo territorio.

8. Función de generación de mapas

```
# ETAPA 7: FUNCION PARA CREAR MAPAS INTERACTIVOS
crear_mapa_producto <- function(datos, producto, zoom = 7) {
  pal <- colorNumeric("YlGn", domain = datos[[producto]],
    na.color = "gray90")

  (datos) %>%
    addTiles() %>%
    addCircleMarkers(
      lng = ~LONGITUD,
      lat = ~LATITUD,
      radius = ~sqrt(get(producto)),
      color = "darkgreen",
      fillColor = ~pal(get(producto)),
      fillOpacity = 0.7,
      popup = ~paste(NOMBREDI, "<br>", producto, ":",
        round(get(producto),1), "ton")
    ) %>%
    addLegend("bottomright", pal = pal, values = ~get(producto),
      title = paste("Produccion", producto, "(ton)")) %>%
    setView(lng = -70.02, lat = -15.84, zoom = zoom)
}
```

Interpretación: Se define una función modular para crear mapas interactivos de cualquier producto agrícola. Los elementos clave son:

Se utiliza una escala verde-amarilla (YlGn) que asigna tonalidades según la magnitud de producción. El radio de cada círculo es proporcional a la raíz cuadrada de la producción, evitando que valores extremos dominen visualmente el mapa. Al hacer clic sobre un marcador, se despliega el nombre del distrito y su producción redondeada a un decimal. El mapa se inicializa en las coordenadas del lago Titicaca (−15,84, −70,02) con un nivel de zoom predeterminado de 7.

9. Generación de mapas por producto

```
# ETAPA 8: CREAR MAPAS
mapa_maiz <- crear_mapa_producto(ena_wide, "MAIZ AMARILLO DURO")
mapa_quinoa <- crear_mapa_producto(ena_wide, "QUINUA")
mapa_alfalfa <- crear_mapa_producto(ena_wide, "ALFALFA")
```

Interpretación: Se instancian tres objetos de mapa independientes, uno por cada producto analizado. Cada mapa hereda las propiedades definidas en la función `crear_mapa_producto`, permitiendo la visualización comparativa de patrones espaciales de producción.

10. Exportación de mapas interactivos

```
# ETAPA 9: GUARDAR MAPAS INTERACTIVOS
saveWidget(mapa_maiz, "mapa_maiz_puno.html", selfcontained = TRUE)
saveWidget(mapa_quinoa, "mapa_quinoa_puno.html", selfcontained = TRUE)
saveWidget(mapa_alfalfa, "mapa_alfalfa_puno.html", selfcontained = TRUE)
```

Interpretación: Los mapas interactivos se guardan como archivos HTML autónomos mediante la función `saveWidget`. El argumento `selfcontained = TRUE` asegura que todas las dependencias de JavaScript y CSS se embeben en el archivo resultante, permitiendo su visualización offline sin requerir conexión a internet ni servidores externos.

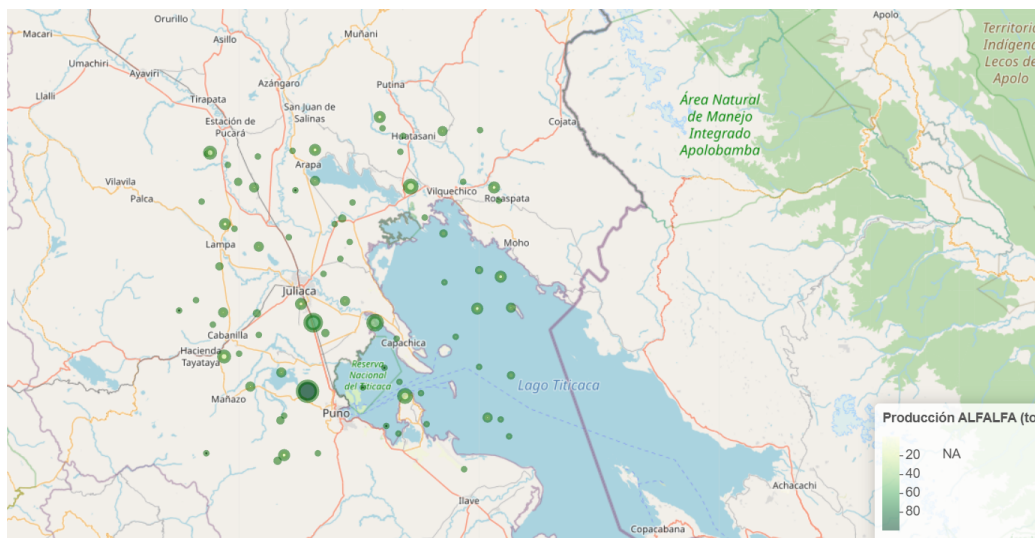


Figura 1: Ejemplo de visualización interactiva de producción de ALFALFA generada. Los marcadores circulares representan distritos, con tamaño proporcional a la producción y color según escala numérica.

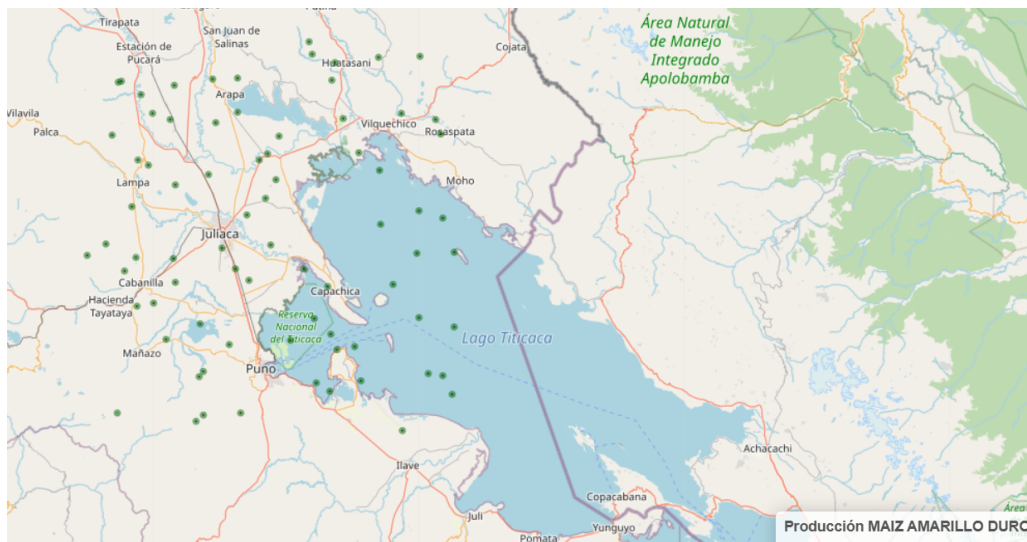


Figura 2: Ejemplo de visualización interactiva de producción DE MAIZ AMARILLO . Los marcadores circulares representan distritos, con tamaño proporcional a la producción y color según escala numérica.

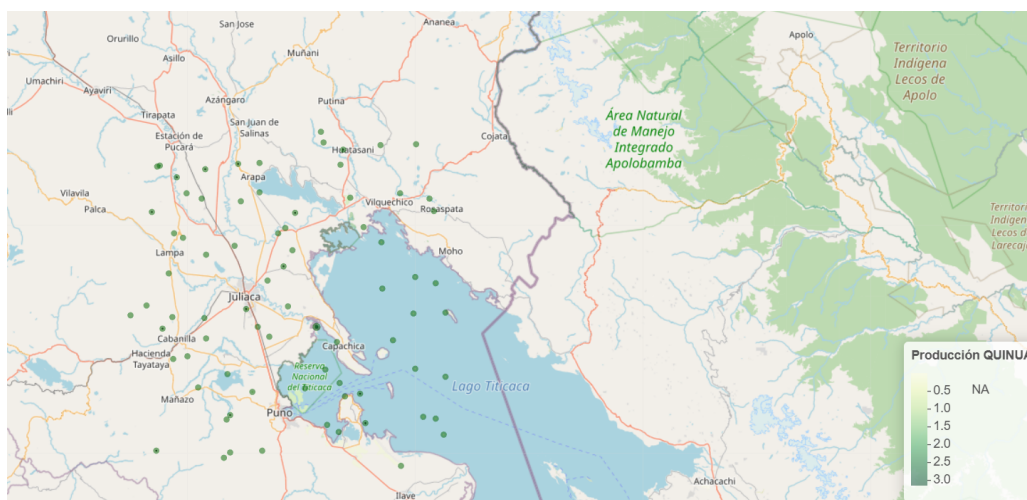


Figura 3: Ejemplo de visualización interactiva de producción de QUINUA . Los marcadores circulares representan distritos, con tamaño proporcional a la producción y color según escala numérica.

Conclusiones

Este flujo de trabajo permite analizar espacialmente la producción agrícola del departamento de Puno mediante visualizaciones interactivas. Las principales ventajas metodológicas son:

La función `crear_mapa_producto` puede reutilizarse para cualquier producto en el dataset ENA. El uso de raíz cuadrada para el radio de los marcadores evita la saturación visual en regiones de alta producción. Los archivos HTML generados son completamente autónomos y pueden compartirse fácilmente. El uso de `set.seed` garantiza que las coordenadas simuladas sean idénticas en ejecuciones subsecuentes.

Como trabajo futuro, se recomienda incorporar shapefiles oficiales del INEI para reemplazar las coordenadas simuladas por centroides geométricos reales, mejorando así la precisión espacial del análisis.