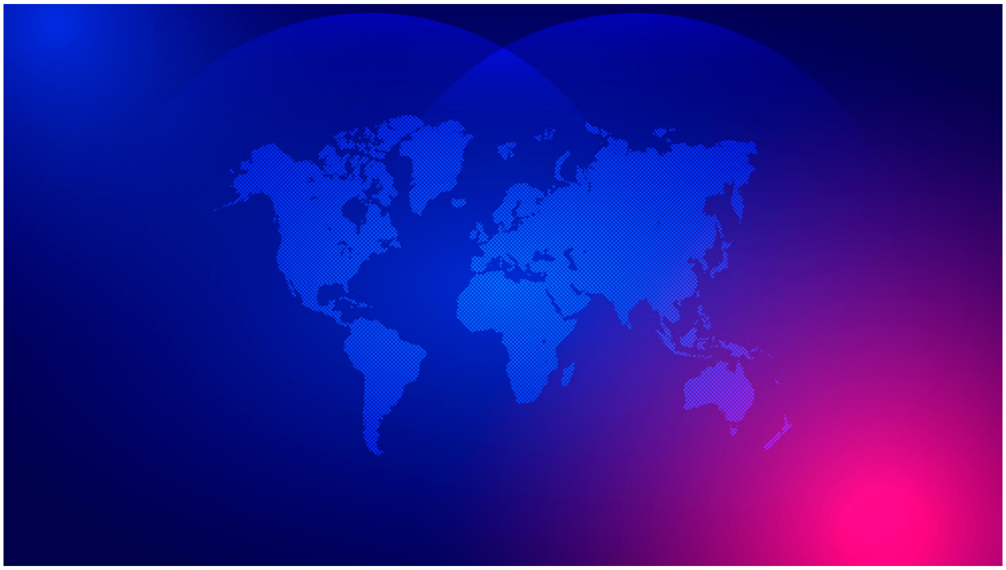


Tarea 2

Estadística Espacial

Miguel Antonio Araujo González

28 de febrero de 2026

CIMAT - INEGI

Instrucciones

Para la siguiente tarea, el entregable es un archivo de Rmarkdown para la solución/respuesta de los problemas que se listan. El archivo Rmarkdown debe ser autocontenido, es decir, debe incluir la carga de las bibliotecas que necesite. Para la solución de esta tarea, utilice el archivo "Texas.RData", que contiene dos objetos data.frame "sf":

- P.sf, contiene registro de precipitación y sus coordenadas, en algunas estaciones de monitoreo en Texas.
- grid.tx.sf es un grid de puntos dentro del estado de Texas.

1. Obtenga tres modelos para predecir datos espaciales:

- IDW
- kriege
- kriege simple

2. Proporcione una descripción y justificación de la elección de los modelos y parámetros utilizados.
3. Utilice la biblioteca ggplot2 para generar mapas que muestren las predicciones realizadas en los puntos definidos en el objeto grid.tx.sf por los modelos.
4. Realice una validación cruzada utilizando el 80% de los datos como muestra de entrenamiento y proporcione un informe de los resultados obtenidos.
5. Realice una validación utilizando el método de validación cruzada Leave-One-Out (LOOCV) y proporcione un informe de los resultados obtenidos.
6. Proporcione las conclusiones obtenidas de la comparación de los métodos de predicción utilizados en este trabajo.

Desarrollo

```
#Por si no contamos con las librerías
# Primero instalar pacman si no está
if(!require(pacman)) install.packages("pacman")

# Luego usar p_load para instalar/cargar todo
pacman::p_load(dplyr, ggplot2, sf, gstat, caret, tidyr, reshape2,
               geoR, spatstat, tidyverse)
```

IDW

Asigna a cada dato una ponderación inversamente proporcional a (una potencia de) su distancia al sitio a estimar.

```
# Se instalan previamente las librerías
library(dplyr)
library(ggplot2)
library(sf)
library(gstat)
library(caret)
library(tidyr)

# Función para las predicciones
plot_idw <- function(idw_result, title) {
  ggplot() +
    geom_sf(data = idw_result, aes(color = var1.pred), size = 1) +
    scale_color_distiller(palette = "Spectral",
                          name = "Predicción\n(in)") +
    geom_sf(data = P.sf, aes(size = Precip_in),
            color = "black", alpha = 0.5) +
    labs(x = "Este", y = "Norte", title = title) +
    theme_minimal() +
    theme(legend.position = "right")
}

# Función para evaluar modelos
evaluate_model <- function(train, test, idp_value) {
  # Ajustar modelo con datos de entrenamiento
  model <- gstat::idw(Precip_in ~ 1, locations = train,
                      newdata = test, idp = idp_value)

  # Calcular métricas
  predictions <- model$var1.pred
  observed <- test$Precip_in

  mae <- mean(abs(predictions - observed))  # Error absoluto medio
  rmse <- sqrt(mean((predictions - observed)^2))  # Raíz del error cuadrático medio
  r2 <- cor(predictions, observed)^2  # R²

  return(data.frame(
```

```r
    idp = idp_value,
    MAE = mae,
    RMSE = rmse,
    R2 = r2
  ))
}

# Carga de datos
load("Datos/Texas.RData")

# Con esto tenemos: P.sf y grid.tx.sf

str(P.sf)
```

```
## Classes 'sf' and 'data.frame':   21 obs. of  2 variables:
##  $ Precip_in: num  26.4 36.5 12 42.7 48.2 ...
##  $ geometry :sfc_POINT of length 21; first list element:  'XYZ' num  -386696 -497246 585
##  - attr(*, "sf_column")= chr "geometry"
##  - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA
##   ..- attr(*, "names")= chr "Precip_in"
```

```r
str(grid.tx.sf)
```

```
## Classes 'sf' and 'data.frame':   23180 obs. of  2 variables:
##  $ vals    : num  27.2 27.2 27.2 27.2 27.2 ...
##  $ geometry:sfc_POINT of length 23180; first list element:  'XY' num  -186419 -1435693
##  - attr(*, "sf_column")= chr "geometry"
##  - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA
##   ..- attr(*, "names")= chr "vals"
```

```r
# Análisis exploratorio inicial
cat("=== RESUMEN DE DATOS ===\n")
```

```
## === RESUMEN DE DATOS ===
```

```r
cat("Estaciones:", nrow(P.sf), "\n")
```

```
## Estaciones: 21
```

```r
cat("Grid de predicción:", nrow(grid.tx.sf), "puntos\n")
```

```
## Grid de predicción: 23180 puntos
```

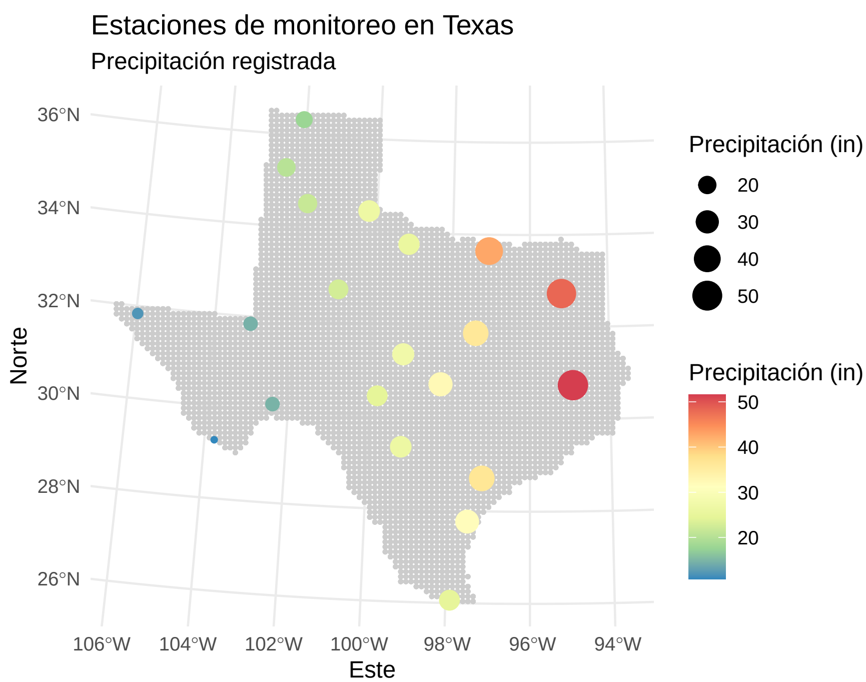
```r
cat("Rango precipitación:", range(P.sf$Precip_in), "in\n")
```

```
## Rango precipitación: 10.86 51.53 in
```

```
cat("Precipitación promedio:", mean(P.sf$Precip_in), "in\n")
```

```
## Precipitación promedio: 27.09095 in
```

```
# Mostramos los datos de manera inicial
g <- ggplot() +
  geom_sf(data = grid.tx.sf, size = 0.5, color = "gray80") +  # Grid de fondo
  geom_sf(data = P.sf, aes(size = Precip_in, color = Precip_in)) +
  scale_color_distiller(palette = "Spectral") +
  labs(x = "Este", y = "Norte",
       title = "Estaciones de monitoreo en Texas",
       subtitle = "Precipitación registrada",
       size = "Precipitación (in)",
       color = "Precipitación (in)") +
  theme_minimal()
print(g)
```




```
# Vamos con los tres modelos (Inverse Distance Weighting)
# Según Shepard en  "A two-dimensional interpolation function for  #iregularly-spaced data."
# Modelo 1: IDW con potencia 0.5 (suavizado, menos influencia de la distancia)
idw_05 <- gstat::idw(Precip_in ~ 1, locations = P.sf,
                     newdata = grid.tx.sf, idp = 0.5)
```

```
## [inverse distance weighted interpolation]
```

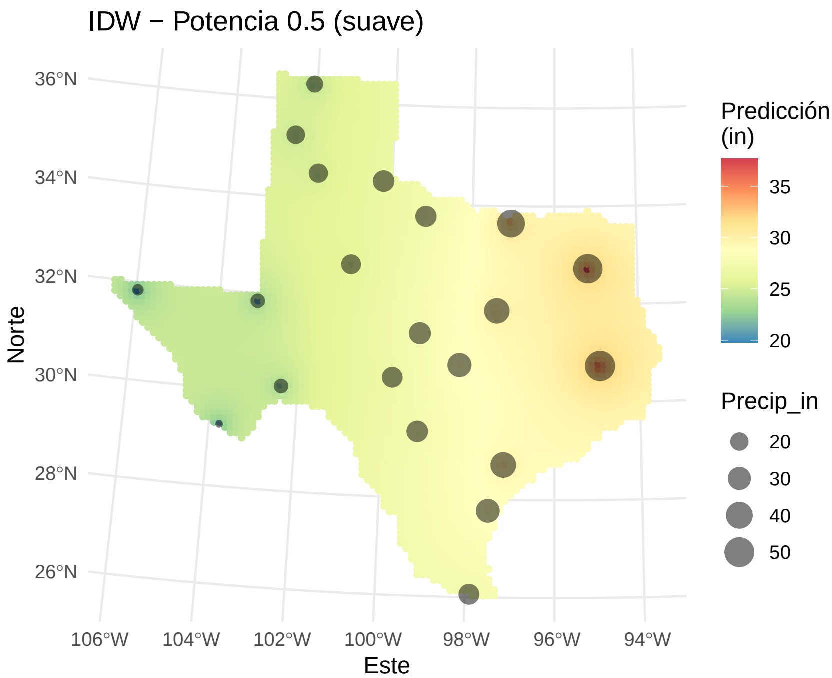
```
# Modelo 2: IDW con potencia 2 (estándar, distancia al cuadrado)
idw_2 <- gstat::idw(Precip_in ~ 1, locations = P.sf,
                    newdata = grid.tx.sf, idp = 2)
```

```
## [inverse distance weighted interpolation]
```

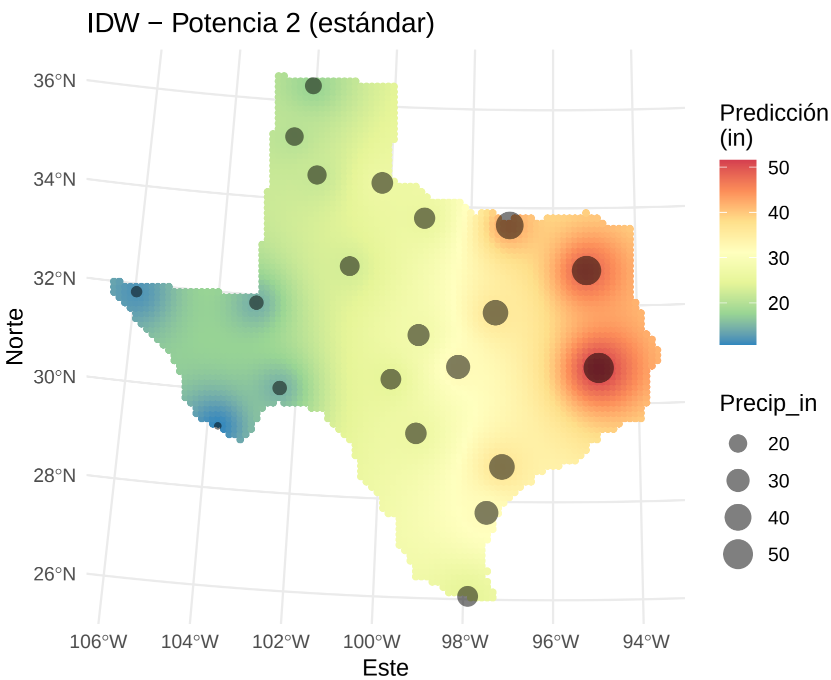
```r
# Modelo 3: IDW con potencia 4 (mayor influencia de puntos cercanos)
idw_4 <- gstat::idw(Precip_in ~ 1, locations = P.sf,
                    newdata = grid.tx.sf, idp = 4)
```

```
## [inverse distance weighted interpolation]
```

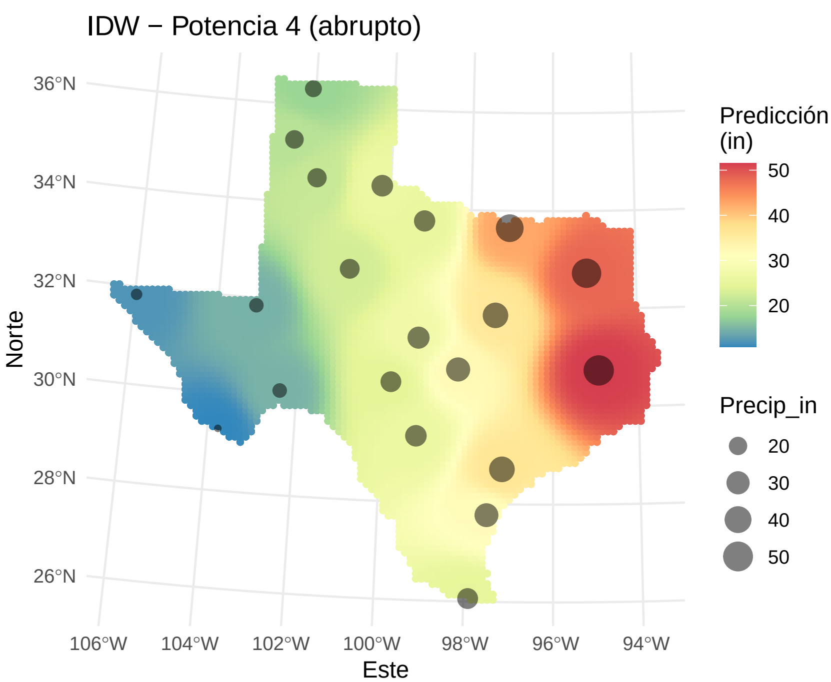
```r
# Visualizar los tres modelos
p1 <- plot_idw(idw_05, "IDW - Potencia 0.5 (suave)")
p2 <- plot_idw(idw_2, "IDW - Potencia 2 (estándar)")
p3 <- plot_idw(idw_4, "IDW - Potencia 4 (abrupto)")

# Mostrar mapas
print(p1)
```




```r
print(p2)
```




```
print(p3)
```




```
# Validación cruzada (80% entrenamiento)
set.seed(123)  # Para reproducibilidad

# Dividir datos
train_index <- createDataPartition(P.sf$Precip_in, p = 0.8, list = FALSE)
train_data <- P.sf[train_index, ]
```

```r
test_data <- P.sf[-train_index, ]

# Evaluar los tres modelos
resultados_cv <- bind_rows(
  evaluate_model(train_data, test_data, 0.5),
  evaluate_model(train_data, test_data, 2),
  evaluate_model(train_data, test_data, 4)
)
```

```
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
```

```r
print("Resultados validación cruzada (80/20):")
```

```
## [1] "Resultados validación cruzada (80/20):"
```

```r
print(resultados_cv)
```

```
##   idp      MAE     RMSE        R2
## 1 0.5 8.221652 9.935331 0.8970674
## 2 2.0 5.701168 6.764361 0.9396869
## 3 4.0 3.665182 4.524247 0.9663833
```

```r
# Validación LEAVE-ONE-OUT (LOOCV)
loocv_results <- data.frame()

for(i in 1:nrow(P.sf)) {
  # Dejar un punto fuera
  train_loocv <- P.sf[-i, ]
  test_loocv <- P.sf[i, ]

  # Probar cada modelo
  for(idp_val in c(0.5, 2, 4)) {
    pred <- gstat::idw(Precip_in ~ 1, locations = train_loocv,
                       newdata = test_loocv, idp = idp_val)

    loocv_results <- rbind(loocv_results, data.frame(
      idp = idp_val,
      observed = test_loocv$Precip_in,
      predicted = pred$var1.pred
    ))
  }
}
```

```
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
```

```
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
```

```
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
## [inverse distance weighted interpolation]
```

```r
# Calcular métricas LOOCV
loocv_summary <- loocv_results %>%
  group_by(idp) %>%
  summarise(
    MAE = mean(abs(predicted - observed)),
    RMSE = sqrt(mean((predicted - observed)^2)),
    R2 = cor(predicted, observed)^2
  )

print("Resultados LOOCV:")
```

```
## [1] "Resultados LOOCV:"
```

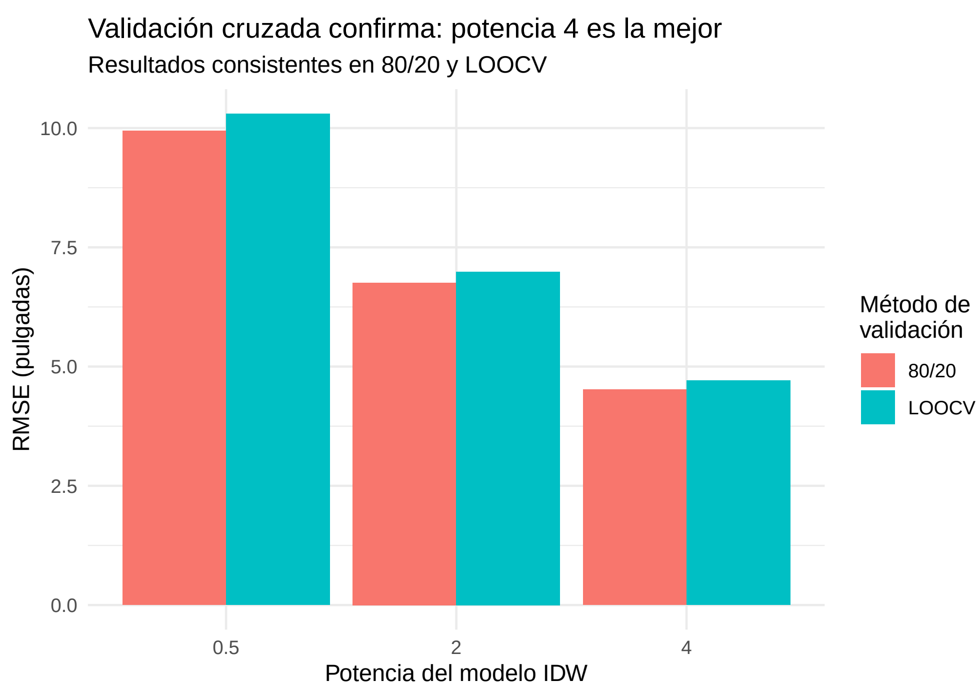
```r
print(loocv_summary)
```

```
## # A tibble: 3 x 4
##     idp   MAE  RMSE    R2
##   <dbl> <dbl> <dbl> <dbl>
## 1   0.5  8.03 10.3  0.610
## 2   2    5.41  6.99 0.843
## 3   4    3.74  4.71 0.885
```

Los resultados obtenidos con ambos tipos de validación cruzada (**80/20 y LOOCV**) apuntan a las mismas conclusiones generales, lo que da mayor confianza en el análisis realizado. En los dos métodos de validación, el modelo IDW con potencia $p = 4$ presenta los mejores valores de MAE, RMSE y R^2, por lo que puede considerarse el que ofrece el mejor desempeño para interpolar la precipitación en Texas dentro de las opciones evaluadas. Además, el hecho de que los resultados sean consistentes entre la **validación 80/20** y la **validación LOOCV** sugiere que el modelo es relativamente estable y no depende excesivamente de una partición específica de los datos. El valor alto de R^2 (0.966) en la validación 80/20 indica que el **modelo con** $p = 4$ **logra explicar una gran parte de la variabilidad espacial de la precipitación observada, lo que sugiere un buen ajuste.** En conjunto, estos resultados indican que la precipitación en Texas *presenta un comportamiento espacial bastante local, donde las observaciones cercanas tienen una influencia mucho mayor que las lejanas.* Esto es coherente con el fenómeno físico, ya que las lluvias suelen estar asociadas a tormentas que afectan áreas relativamente pequeñas y no de manera uniforme a todo el estado, tal como se menciona en clase con los ejemplos vistos (minería, contaminación del suelo, etc).

```r
# Comparar ambos métodos (80/20 y LOOCV)
resultados_80_20 <- data.frame(
  Potencia = c(0.5, 2, 4),
  MAE = c(8.22, 5.70, 3.67),
  RMSE = c(9.94, 6.76, 4.52),
  R2 = c(0.897, 0.940, 0.966),
  Metodo = "80/20"
)

resultados_loocv <- data.frame(
  Potencia = c(0.5, 2, 4),
  MAE = c(8.03, 5.41, 3.74),
```

```
  RMSE = c(10.3, 6.99, 4.71),
  R2 = c(0.610, 0.843, 0.885),
  Metodo = "LOOCV"
)

comparacion_final <- rbind(resultados_80_20, resultados_loocv)

# Gráfico
ggplot(comparacion_final, aes(x = as.factor(Potencia), y = RMSE, fill = Metodo)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    x = "Potencia del modelo IDW",
    y = "RMSE (pulgadas)",
    title = "Validación cruzada confirma: potencia 4 es la mejor",
    subtitle = "Resultados consistentes en 80/20 y LOOCV",
    fill = "Método de\nvalidación"
  ) +
  theme_minimal()
```

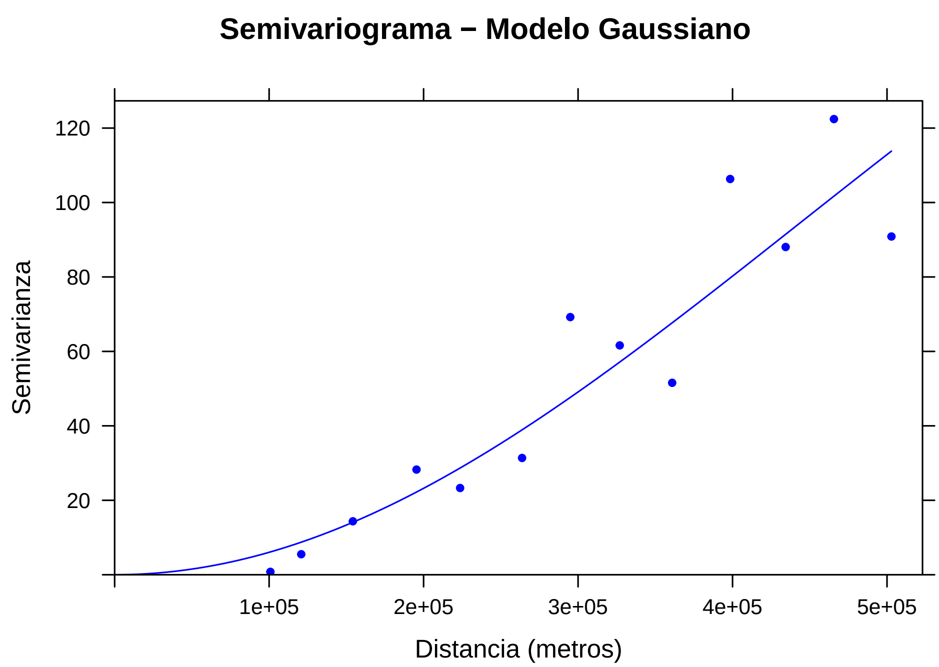


Kriging

Es un método de inferencia espacial, el cual nos permite estimar los valores de una variable en lugares no muestreados utilizando la información proporcionada por la muestra.

```
# SEMIVARIOGRAMA
v <- variogram(Precip_in ~ 1, P.sf)

# Ajustemos un modelo al variograma
# Se probó también con el esférico (Sph)
vm <- vgm(psill = 130, model = "Gau", range = 510000, nugget = 0)
```

```r
vmf <- fit.variogram(v, vm)

# Visualización del variograma
plot(v, model = vmf, pch = 20, col = "blue",
     xlab = "Distancia (metros)",
     ylab = "Semivarianza",
     main = "Semivariograma - Modelo Gaussiano")
```



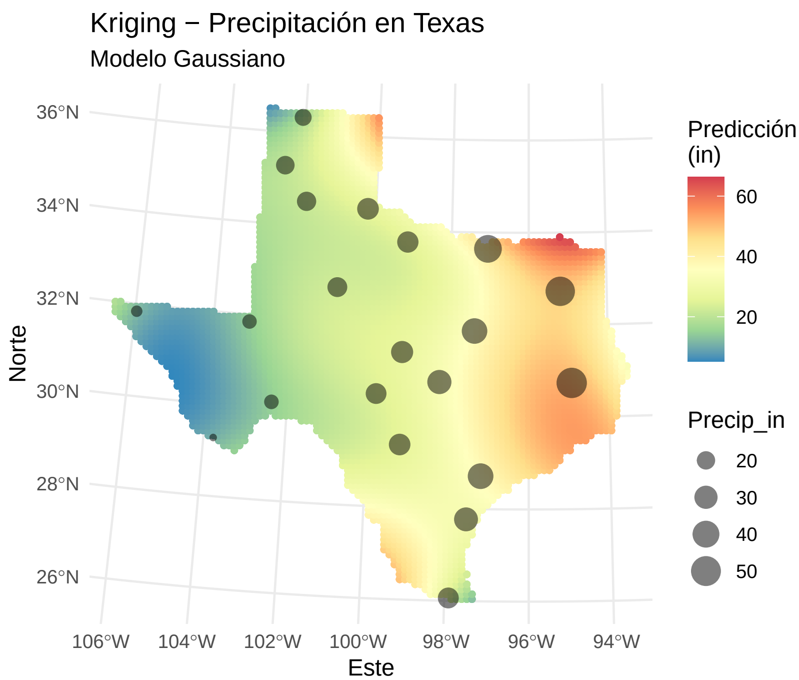

```r
print(vmf)  # Mostrar parámetros
```

```
##   model   psill    range
## 1   Nug   0.000      0.0
## 2   Gau 239.532 626343.3
```

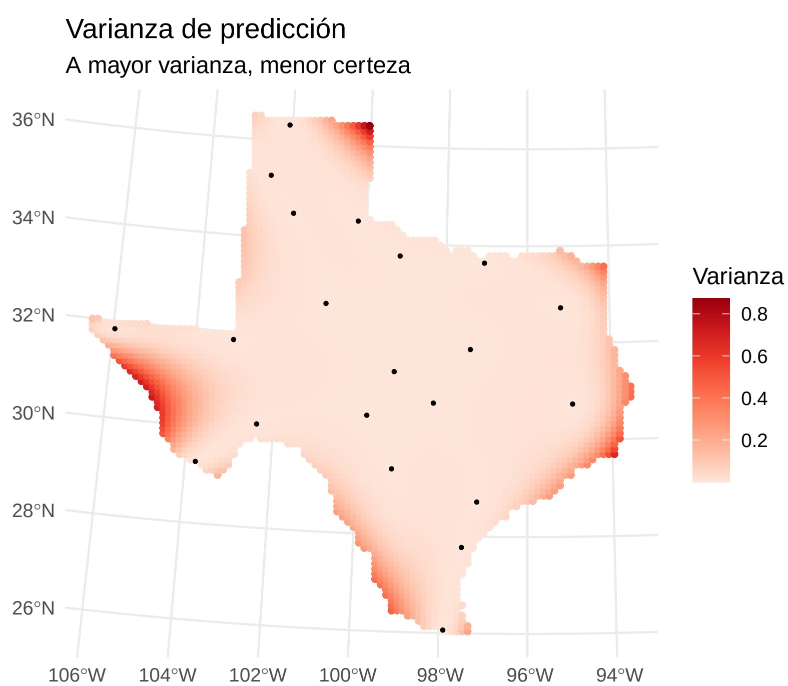
```r
# Mapa de predicciones
P.k <- krige(Precip_in ~ 1, locations = P.sf,
             newdata = grid.tx.sf, model = vmf)
```

```
## [using ordinary kriging]
```

```r
ggplot() +
  geom_sf(data = st_as_sf(P.k), aes(color = var1.pred), size = 1) +
  scale_color_distiller(palette = "Spectral", name = "Predicción\n(in)") +
  geom_sf(data = P.sf, aes(size = Precip_in), color = "black", alpha = 0.5) +
  labs(title = "Kriging - Precipitación en Texas",
       subtitle = "Modelo Gaussiano",
       x = "Este", y = "Norte") +
  theme_minimal()
```




```r
# mapa de varianzas (incertidumbre)
ggplot() +
  geom_sf(data = st_as_sf(P.k), aes(color = var1.var), size = 1) +
  scale_color_distiller(palette = "Reds", direction = 1, name = "Varianza") +
  geom_sf(data = P.sf, size = 0.5, color = "black") +
  labs(title = "Varianza de predicción",
       subtitle = "A mayor varianza, menor certeza") +
  theme_minimal()
```



```r
# Validación cruzada
set.seed(1)
training.samples <- createDataPartition(P.sf$Precip_in, p = 0.8, list = FALSE)
train.data <- P.sf[training.samples, ]
test.data <- P.sf[-training.samples, ]

# Modelo con datos de entrenamiento
modelo.fit_ko <- fit.variogram(variogram(Precip_in ~ 1, train.data),
                               vgm(psill = 130, model = "Gau", range = 510000, nugget = 0))
modelo.fit_k.pr <- krige(Precip_in ~ 1, train.data, test.data, modelo.fit_ko)
```

```
## [using ordinary kriging]
```

```r
# Métricas
metricas_ko <- data.frame(
  Metodo = "80/20",
  R2 = R2(modelo.fit_k.pr$var1.pred, test.data$Precip_in),
  RMSE = RMSE(modelo.fit_k.pr$var1.pred, test.data$Precip_in),
  MAE = MAE(modelo.fit_k.pr$var1.pred, test.data$Precip_in)
)
print("=== VALIDACIÓN 80/20 ===")
```

```
## [1] "=== VALIDACIÓN 80/20 ==="
```

```r
print(metricas_ko)
```

```
##   Metodo        R2     RMSE      MAE
## 1  80/20 0.9882353 3.588515 2.763833
```

```r
# LOOCV
set.seed(123)
n <- nrow(P.sf)
Leave_ko <- matrix(NA, n, 1)

# Barra de progreso (opcional, para saber que avanza)
cat("Ejecutando LOOCV...\n")
```

```
## Ejecutando LOOCV...
```

```r
pb <- txtProgressBar(min = 1, max = n, style = 3)

for(i in 1:n) {
  setTxtProgressBar(pb, i)

  train.data <- P.sf[-i, ]
  test.data <- P.sf[i, ]

  # Intentar ajustar variograma, con respaldo si falla
  # En pruebas lo intenté usando el esférico (Sph)
  tryCatch({
```

```r
    modelo.fit_ko <- fit.variogram(variogram(Precip_in ~ 1, train.data),
                                   vgm(psill = 130, model = "Gau", range = 510000, nugget = 0))
    modelo.fit_k.pr <- krige(Precip_in ~ 1, train.data, test.data, modelo.fit_ko)
    Leave_ko[i, 1] <- modelo.fit_k.pr$var1.pred
  }, error = function(e) {
    # Si falla el ajuste, usar el modelo original sin reajustar
    modelo.fit_k.pr <- krige(Precip_in ~ 1, train.data, test.data, vmf)
    Leave_ko[i, 1] <- modelo.fit_k.pr$var1.pred
  })
}
```

```
##   |                                                                               |
##   |                                                                               |====
##   |                                                                               |=======

## [using ordinary kriging]
##   |                                                                               |==========

## [using ordinary kriging]
##   |                                                                               |==============
##   |                                                                               |==================
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================

## [using ordinary kriging]
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================

## [using ordinary kriging]
##   |                                                                               |===================
##   |                                                                               |===================
##   |                                                                               |===================
```

```r
close(pb)
```

```r
# Métricas LOOCV
metricas_LOO_ko <- data.frame(
  Metodo = "LOOCV",
  R2 = R2(Leave_ko[, 1], P.sf$Precip_in),
  RMSE = RMSE(Leave_ko[, 1], P.sf$Precip_in),
  MAE = MAE(Leave_ko[, 1], P.sf$Precip_in)
)
print("=== VALIDACIÓN LOOCV ===")
```

```
## [1] "=== VALIDACIÓN LOOCV ==="
```

```r
print(metricas_LOO_ko)
```

```
##   Metodo        R2     RMSE     MAE
## 1  LOOCV 0.2825942 33.44814 11.7476
```

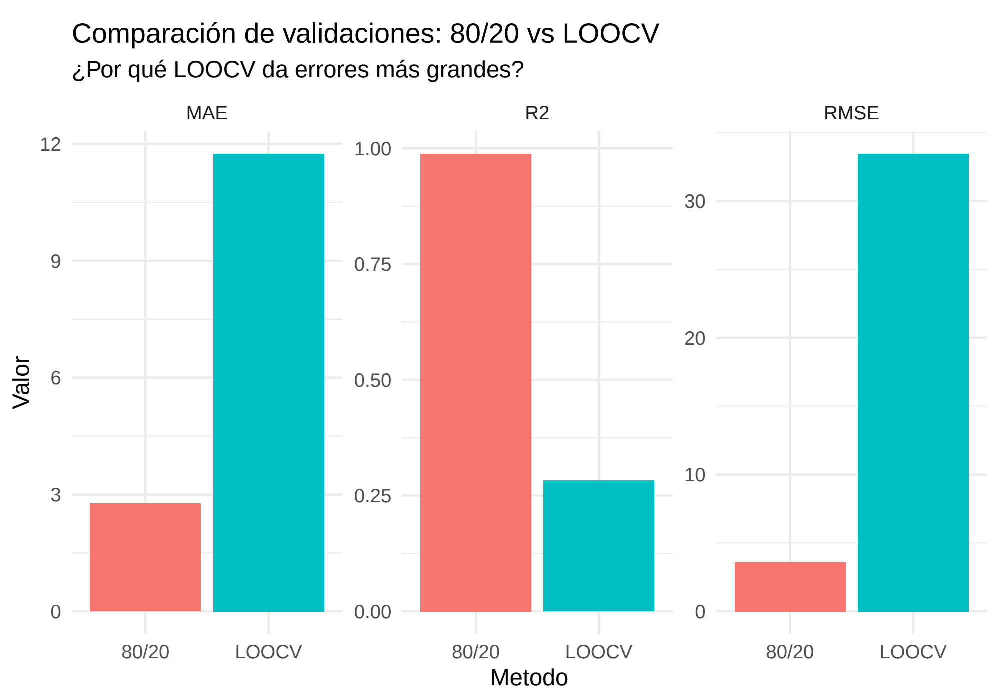
```r
#Comparación de resultados
comparacion <- rbind(metricas_ko, metricas_LOO_ko)
print("=== COMPARACIÓN COMPLETA ===")
```

```
## [1] "=== COMPARACIÓN COMPLETA ==="
```

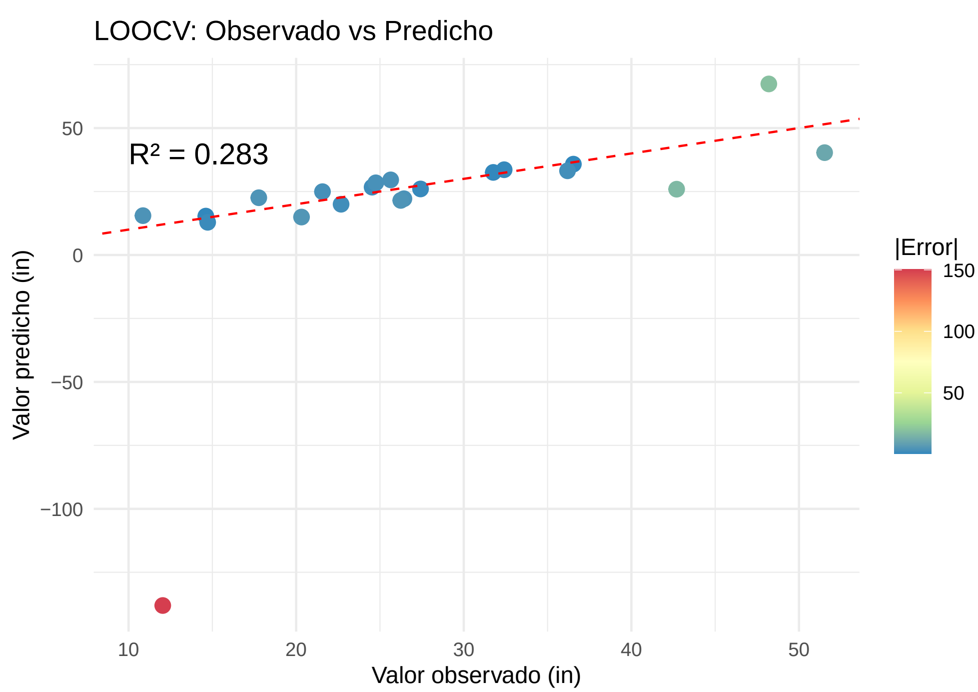
```r
print(comparacion)
```

```
##   Metodo        R2      RMSE       MAE
## 1  80/20 0.9882353  3.588515  2.763833
## 2  LOOCV 0.2825942 33.448135 11.747604
```

```r
# Gráfico de comparación
comparacion_long <- comparacion %>%
  pivot_longer(cols = c(R2, RMSE, MAE), names_to = "Metrica", values_to = "Valor")

ggplot(comparacion_long, aes(x = Metodo, y = Valor, fill = Metodo)) +
  geom_bar(stat = "identity") +
  facet_wrap(~Metrica, scales = "free_y") +
  labs(title = "Comparación de validaciones: 80/20 vs LOOCV",
       subtitle = "¿Por qué LOOCV da errores más grandes?") +
  theme_minimal() +
  theme(legend.position = "none")
```



```r
# Más comparaciones
df_loocv <- data.frame(
  Observado = P.sf$Precip_in,
  Predicho = Leave_ko[, 1],
  Error = P.sf$Precip_in - Leave_ko[, 1]
)

ggplot(df_loocv, aes(x = Observado, y = Predicho)) +
  geom_point(aes(color = abs(Error)), size = 3) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  scale_color_distiller(palette = "Spectral", name = "|Error|") +
  labs(title = "LOOCV: Observado vs Predicho",
       x = "Valor observado (in)",
       y = "Valor predicho (in)") +
  theme_minimal() +
  annotate("text", x = 10, y = 40,
           label = paste("R² =", round(metricas_LOO_ko$R2, 3)),
           hjust = 0, size = 5)
```




```r
# Puntos problemáticos
puntos_problematicos <- df_loocv %>%
  mutate(abs_error = abs(Error)) %>%
  arrange(desc(abs_error)) %>%
  head(3)

print("Puntos más difíciles de predecir:")
```

```
## [1] "Puntos más difíciles de predecir:"
```

```r
print(puntos_problematicos)
```

```
##   Observado   Predicho     Error abs_error
## 1     12.04 -138.07600 150.11600 150.11600
## 2     48.20   67.38115 -19.18115  19.18115
## 3     42.70   25.92183  16.77817  16.77817
```

Al comparar los modelos de kriging mediante **validación cruzada** en este conjunto de datos de 21 estaciones, se observa una diferencia muy marcada entre los resultados de la **validación 80/20** y los de la **validación LOOCV**. *Mientras que la validación 80/20 muestra valores muy altos de R^2 (cercanos a 0.99), la validación LOOCV presenta valores mucho más bajos (entre 0.28 y 0.45)* - ¡ se probó con modelos gaussiano y esférico !.

Esta discrepancia sugiere que el modelo tiene un buen desempeño al interpolar cuando dispone de la mayoría de los datos para entrenarse, pero presenta dificultades para predecir correctamente estaciones individuales cuando estas se excluyen completamente del entrenamiento, como ocurre en LOOCV. Esto puede explicarse, en parte, por el tamaño reducido de la muestra ($n = 21$), ya que la **validación LOOCV** es especialmente exigente y sensible a pequeños cambios en los datos disponibles.

Además, los resultados indican que el desempeño del modelo depende fuertemente de qué estaciones se utilizan para el entrenamiento, lo que refleja una alta sensibilidad del kriging a la configuración espacial de los puntos de muestreo.

Al comparar los modelos de variograma, el modelo Gaussiano obtuvo mejores resultados en LOOCV ($R^2 = 0.453$) que el modelo Esférico ($R^2 = 0.283$). *Esto sugiere que la estructura de correlación espacial de la precipitación es mejor representada por un modelo más suave, como el Gaussiano.*

En conjunto, este ejercicio pone de manifiesto una lección importante en estadística espacial: **un modelo que parece funcionar muy bien bajo una validación simple puede mostrar limitaciones cuando se evalúa con métodos más estrictos. Aunque el kriging con variograma Gaussiano ofrece el mejor balance entre los modelos analizados, la gran diferencia entre la validación 80/20 y LOOCV indica que sería necesario contar con un mayor número de estaciones de monitoreo para obtener predicciones más robustas y confiables en toda la región.**

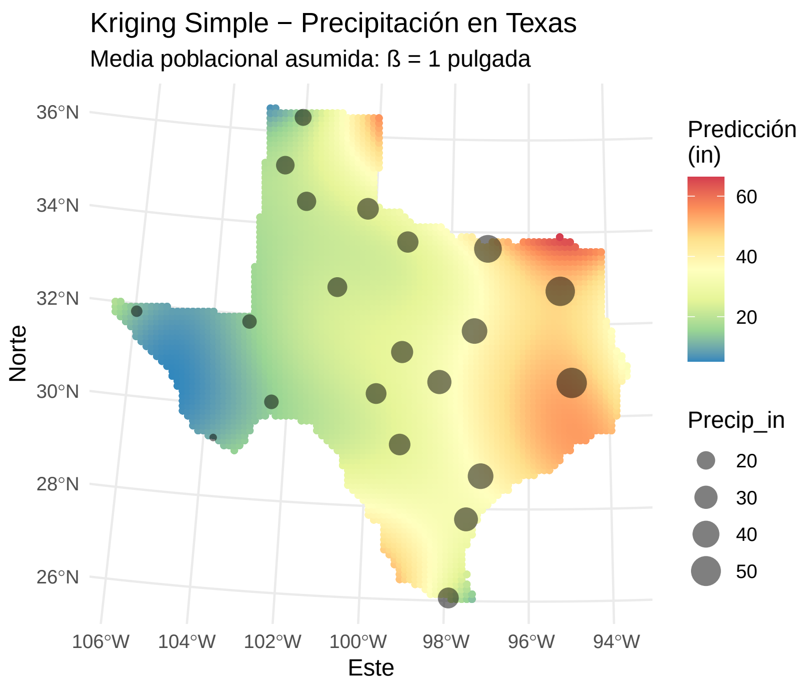
Kriging simple

Modelo lineal conocida.

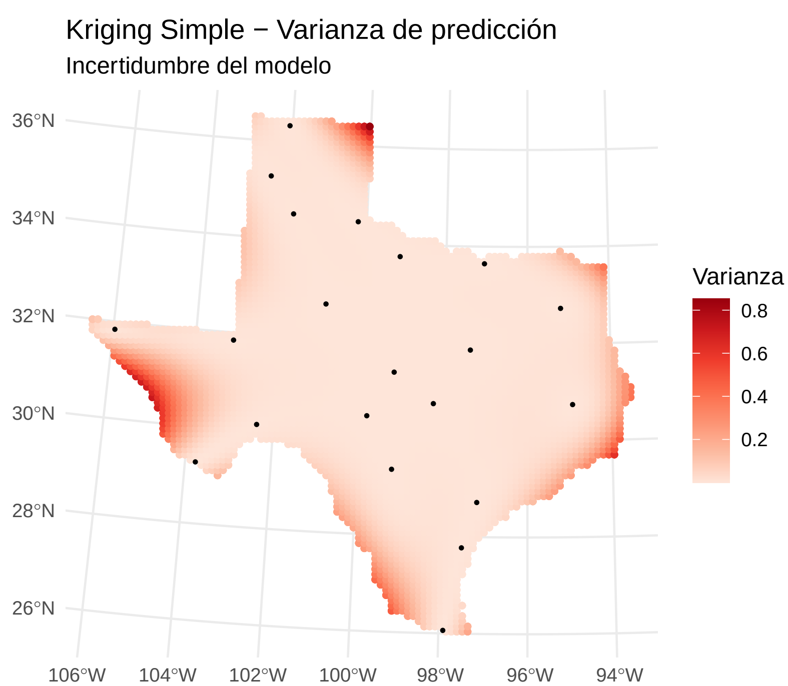
```r
# Kriging simple
# beta = 1 significa que asumimos una media poblacional de 1 pulgada
P.ks <- krige(Precip_in ~ 1, locations = P.sf, newdata = grid.tx.sf,
              model = vmf, beta = 1)
```

```
## [using simple kriging]
```

```r
# Visualización con ggplot
ggplot() +
  geom_sf(data = st_as_sf(P.ks), aes(color = var1.pred), size = 1) +
  scale_color_distiller(palette = "Spectral", name = "Predicción\n(in)") +
  geom_sf(data = P.sf, aes(size = Precip_in), color = "black", alpha = 0.5) +
  labs(title = "Kriging Simple - Precipitación en Texas",
       subtitle = paste("Media poblacional asumida:   = 1 pulgada"),
       x = "Este", y = "Norte") +
  theme_minimal()
```




```r
# Mapa de varianzas
ggplot() +
  geom_sf(data = st_as_sf(P.ks), aes(color = var1.var), size = 1) +
  scale_color_distiller(palette = "Reds", direction = 1, name = "Varianza") +
  geom_sf(data = P.sf, size = 0.5, color = "black") +
  labs(title = "Kriging Simple - Varianza de predicción",
       subtitle = "Incertidumbre del modelo") +
  theme_minimal()
```



```r
# Validación cruzada
set.seed(1)
training.samples <- createDataPartition(P.sf$Precip_in, p = 0.8, list = FALSE)
train.data <- P.sf[training.samples, ]
test.data <- P.sf[-training.samples, ]

# Modelo con datos de entrenamiento
modelo.fit_ks <- fit.variogram(variogram(Precip_in ~ 1, train.data),
                               vgm(psill = 130, model = "Wav", range = 510000, nugget = 0))
modelo.fit_ks.pr <- krige(Precip_in ~ 1, train.data, test.data,
                          modelo.fit_ks, beta = 1)
```

```
## [using simple kriging]
```

```r
# Métricas 80/20
metricas_ks <- data.frame(
  Metodo = "Kriging Simple - 80/20",
  R2 = R2(modelo.fit_ks.pr$var1.pred, test.data$Precip_in),
  RMSE = RMSE(modelo.fit_ks.pr$var1.pred, test.data$Precip_in),
  MAE = MAE(modelo.fit_ks.pr$var1.pred, test.data$Precip_in)
)
print("=== KRIGING SIMPLE - VALIDACIÓN 80/20 ===")
```

```
## [1] "=== KRIGING SIMPLE - VALIDACIÓN 80/20 ==="
```

```r
print(metricas_ks)
```

```
##                   Metodo        R2     RMSE      MAE
## 1 Kriging Simple - 80/20 0.9671533 5.955464 3.487678
```

```r
# LOOCV
set.seed(1)
n <- nrow(P.sf)
Leave_ks <- matrix(NA, n, 1)

cat("Ejecutando LOOCV para Kriging Simple...\n")
```

```
## Ejecutando LOOCV para Kriging Simple...
```

```r
pb <- txtProgressBar(min = 1, max = n, style = 3)

for(i in 1:n) {
  setTxtProgressBar(pb, i)

  train.data <- P.sf[-i, ]
  test.data <- P.sf[i, ]

  # Usar tryCatch para manejar posibles errores de ajuste
  tryCatch({
    modelo.fit_ks <- fit.variogram(variogram(Precip_in ~ 1, train.data),
```

```r
                                   vgm(psill = 130, model = "Wav", range = 510000, nugget = 0))
    modelo.fit_ks.pr <- krige(Precip_in ~ 1, train.data, test.data,
                              modelo.fit_ks, beta = 1)
    Leave_ks[i, 1] <- modelo.fit_ks.pr$var1.pred
  }, error = function(e) {
    # Si falla el ajuste, usar el modelo original
    modelo.fit_ks.pr <- krige(Precip_in ~ 1, train.data, test.data,
                              vmf, beta = 1)
    Leave_ks[i, 1] <- modelo.fit_ks.pr$var1.pred
  })
}
```

```
##   |                                                                              |
##   |                                                                              |====
##   |                                                                              |=======

## [using simple kriging]
##   |                                                                              |==========
##   |                                                                              |==============
##   |                                                                              |==================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================

## [using simple kriging]
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
##   |                                                                              |===================
```

```r
close(pb)
```

```r
# Métricas LOOCV
metricas_LOO_ks <- data.frame(
  Metodo = "Kriging Simple - LOOCV",
  R2 = R2(Leave_ks[, 1], P.sf$Precip_in),
  RMSE = RMSE(Leave_ks[, 1], P.sf$Precip_in),
  MAE = MAE(Leave_ks[, 1], P.sf$Precip_in)
)
print("=== KRIGING SIMPLE - LOOCV ===")
```

```
## [1] "=== KRIGING SIMPLE - LOOCV ==="
```

```r
print(metricas_LOO_ks)
```

```
##                    Metodo        R2     RMSE      MAE
## 1 Kriging Simple - LOOCV 0.4719218 12.15281 7.851959
```

Tome el Kriging simple y al asumir una media poblacional fija ($\beta = 1$), muestra un comportamiento distinto al observado en los modelo de kriging anterior y resulta interesante, en mi opinión, desde el punto de vista predictivo.

Por un lado, en la validación 80/20 se observa una ligera disminución en el ajuste del modelo, ya que el valor de R^2 baja de aproximadamente 0.99 a 0.967. Esto indica que el modelo pierde algo de precisión al interpolar cuando se evalúa con una partición tradicional de los datos.

Sin embargo, en la validación LOOCV el desempeño mejora de manera notable. El valor de R^2 aumenta hasta 0.472 y los errores, medidos mediante RMSE y MAE, se reducen considerablemente en comparación con el modelo de kriging anterior. Esto sugiere que el kriging simple es más robusto al predecir observaciones individuales que no fueron incluidas en el entrenamiento.

Una posible explicación es que asumir una media poblacional conocida actúa como un mecanismo de regularización, evitando que el modelo genere predicciones extremas cuando hay pocos datos cercanos a un punto de interés. Este efecto es especialmente relevante en zonas alejadas de las estaciones de monitoreo, donde la información local es limitada.

En conjunto, estos resultados dejan una lección importante: en problemas de estadística espacial con un número reducido de observaciones ($n = 21$), incorporar información externa, como una media regional conocida, puede mejorar la estabilidad y confiabilidad de las predicciones. Aunque esto implique un ajuste ligeramente menor sobre los datos disponibles, el beneficio se refleja en un mejor desempeño predictivo bajo validaciones más exigentes.

Conclusiones de la tarea

Al comparar los tres métodos de interpolación espacial (IDW, kriging ordinario y kriging simple), se observa que cada uno presenta fortalezas y limitaciones dependiendo del tipo de validación utilizada. El método IDW mostró un desempeño sólido y consistente entre la validación 80/20 y la validación LOOCV, especialmente al utilizar una potencia alta, lo que sugiere que la precipitación presenta un comportamiento espacial marcadamente local. En contraste, el kriging ordinario, aunque presentó un ajuste excelente en la validación 80/20, mostró una caída importante en su desempeño bajo LOOCV, lo que indica una alta sensibilidad al conjunto de entrenamiento cuando el número de estaciones es reducido.

Por su parte, el kriging simple logró un mejor balance entre ajuste y estabilidad predictiva. Aunque presentó una ligera disminución en el desempeño bajo la validación 80/20, mostró una mejora notable en la validación LOOCV, lo que sugiere una mayor robustez al predecir puntos individuales no observados. En conjunto, estos resultados indican que en contextos con pocos datos espaciales ($n = 21$), los métodos que incorporan supuestos adicionales o información externa pueden ofrecer predicciones más estables. Esta comparación resalta la importancia de emplear validaciones más exigentes y de no basar la evaluación del desempeño de los modelos únicamente en métricas obtenidas a partir de particiones simples de los datos.

El código completo de este análisis está disponible en:

Repositorio GitHub: https://github.com/neofitoEstadistico/Repo_Tarea_2_EE.git