

ANALISIS-ESPACIAL-CON-MODELO-GAUSSIANO.R

WILL

2025-10-19

```
#=====
# ANÁLISIS ESPACIAL CON MODELO GAUSSIANO
# Autor: Willy Vilca Apaza
# Código: 214654
# Curso: Estadística Espacial
# Docente: Ing. Torres Cruz Fred
# Fecha: 17 de setiembre de 2025
#=====
# Objetivo: Modelar la dependencia espacial de precios de viviendas
# Dataset: Housing.csv
# Método: Variogramas y Kriging Gaussiano
#=====

# 1. CONFIGURACIÓN DEL ENTORNO =====
rm(list = ls())

suppressPackageStartupMessages({
  library(readr)
  library(dplyr)
  library(ggplot2)
  library(sf)
  library(gstat)
  library(sp)
  library(viridis)
  library(gridExtra)
})

cat("=====\n")
```

```
## =====
```

```
cat("ANÁLISIS ESPACIAL CON MODELO GAUSSIANO\n")
```

```
## ANÁLISIS ESPACIAL CON MODELO GAUSSIANO
```

```
cat("Dataset: Housing Prices\n")
```

```
## Dataset: Housing Prices
```

```
cat("=====\n\n")
```

```
## =====
```

```
# 2. CARGA Y EXPLORACIÓN DE DATOS =====
cat("FASE 1: CARGA Y EXPLORACIÓN DE DATOS\n")
```

```
## FASE 1: CARGA Y EXPLORACIÓN DE DATOS
```

```
cat("-----\n\n")
```

```
## -----
```

```
housing <- read_csv("D:/Estadística e Informática 2025-II/TECNOLOGÍAS EMERGENTES/I UNIDAD/Housing.csv",
                  show_col_types = FALSE)

cat("Estructura del dataset:\n")
```

```
## Estructura del dataset:
```

```
cat(sprintf(" - Observaciones: %d\n", nrow(housing)))
```

```
## - Observaciones: 545
```

```
cat(sprintf(" - Variables: %d\n\n", ncol(housing)))
```

```
## - Variables: 13
```

```
print(head(housing, 3))
```

```
## # A tibble: 3 × 13
##   price area bedrooms bathrooms stories mainroad guestroom basement
##   <dbl> <dbl>   <dbl>     <dbl>   <dbl> <chr>    <chr>    <chr>
## 1 13300000 7420     4         2       3 yes     no       no
## 2 12250000 8960     4         4       4 yes     no       no
## 3 12250000 9960     3         2       2 yes     no       yes
## # i 5 more variables: hotwaterheating <chr>, airconditioning <chr>,
## #   parking <dbl>, prefarea <chr>, furnishingstatus <chr>
```

```
# 3. GENERACIÓN DE COORDENADAS ESPACIALES =====
cat("\nNOTA: El dataset no contiene coordenadas geográficas reales.\n")
```

```
##
## NOTA: El dataset no contiene coordenadas geográficas reales.
```

```
cat("Generando coordenadas espaciales sintéticas basadas en características...\n\n")
```

```
## Generando coordenadas espaciales sintéticas basadas en características...
```

```
# Crear coordenadas sintéticas usando características de las viviendas
set.seed(2025)

housing_spatial <- housing %>%
  filter(!is.na(price), price > 0) %>%
  # Eliminar outliers
  filter(price < quantile(price, 0.99),
         price > quantile(price, 0.01)) %>%
  mutate(
    # Coordenadas base + variación según características
    x = 500000 + area * 10 + rnorm(n(), 0, 5000),
    y = 4000000 + bedrooms * 15000 + bathrooms * 10000 + rnorm(n(), 0, 5000),
    log_precio = log(price)
  ) %>%
  select(price, log_precio, area, bedrooms, bathrooms, x, y)

cat(sprintf("Datos preparados: %d observaciones\n\n", nrow(housing_spatial)))
```

```
## Datos preparados: 533 observaciones
```

```
# 4. ESTADÍSTICAS DESCRIPTIVAS =====
cat("FASE 2: ESTADÍSTICAS DESCRIPTIVAS\n")
```

```
## FASE 2: ESTADÍSTICAS DESCRIPTIVAS
```

```
cat("-----\n\n")
```

```
## -----
```

```
cat("Variable: Precio de Vivienda\n")
```

```
## Variable: Precio de Vivienda
```

```
cat("-----\n")
```

```
## -----
```

```
cat(sprintf("Media:           %.2f\n", mean(housing_spatial$price)))
```

```
## Media:           4718387.04
```

```
cat(sprintf("Mediana:           %.2f\n", median(housing_spatial$price)))
```

```
## Mediana:          4340000.00
```

```
cat(sprintf("Desv. Estándar:    %.2f\n", sd(housing_spatial$price)))
```

```
## Desv. Estándar:    1694235.22
```

```
cat(sprintf("Mínimo:           %.2f\n", min(housing_spatial$price)))
```

```
## Mínimo:           1890000.00
```

```
cat(sprintf("Máximo:           %.2f\n", max(housing_spatial$price)))
```

```
## Máximo:           10150000.00
```

```
cat(sprintf("Coef. Variación:    %.2f%%\n\n",  
            (sd(housing_spatial$price)/mean(housing_spatial$price))*100))
```

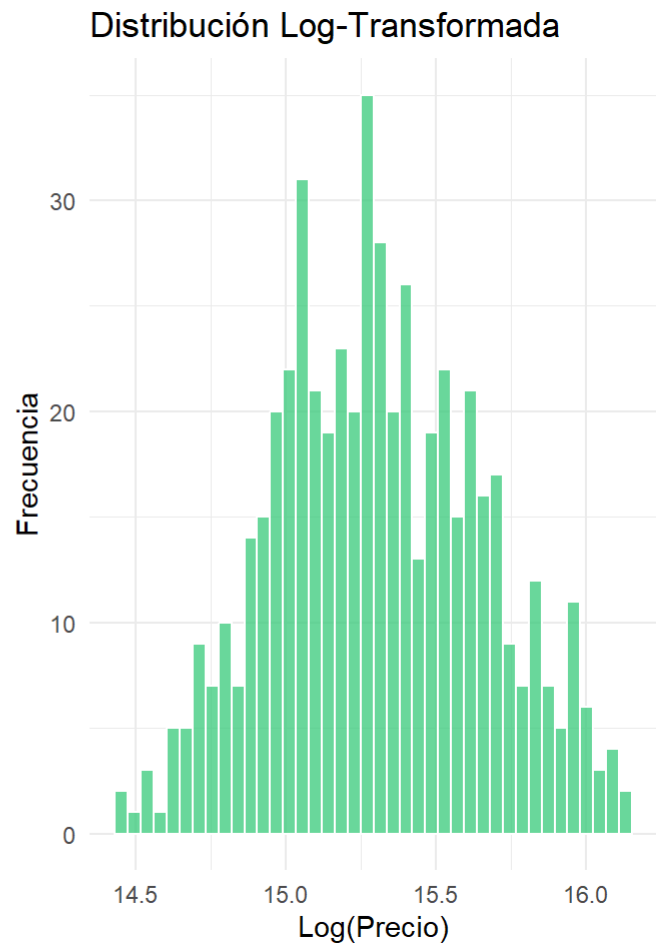
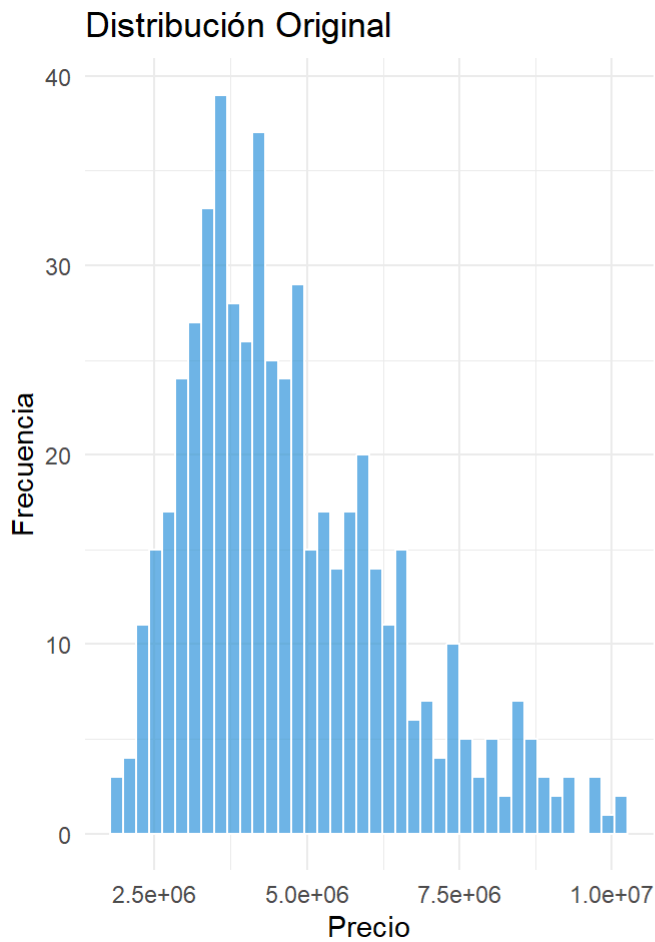
```
## Coef. Variación:    35.91%
```

```
# Histogramas
```

```
p1 <- ggplot(housing_spatial, aes(x = price)) +  
  geom_histogram(bins = 40, fill = "#3498db", alpha = 0.7, color = "white") +  
  labs(title = "Distribución Original",  
        x = "Precio", y = "Frecuencia") +  
  theme_minimal()
```

```
p2 <- ggplot(housing_spatial, aes(x = log_precio)) +  
  geom_histogram(bins = 40, fill = "#2ecc71", alpha = 0.7, color = "white") +  
  labs(title = "Distribución Log-Transformada",  
        x = "Log(Precio)", y = "Frecuencia") +  
  theme_minimal()
```

```
grid.arrange(p1, p2, ncol = 2)
```



```
# 5. VISUALIZACIÓN ESPACIAL =====
cat("FASE 3: VISUALIZACIÓN ESPACIAL\n")
```

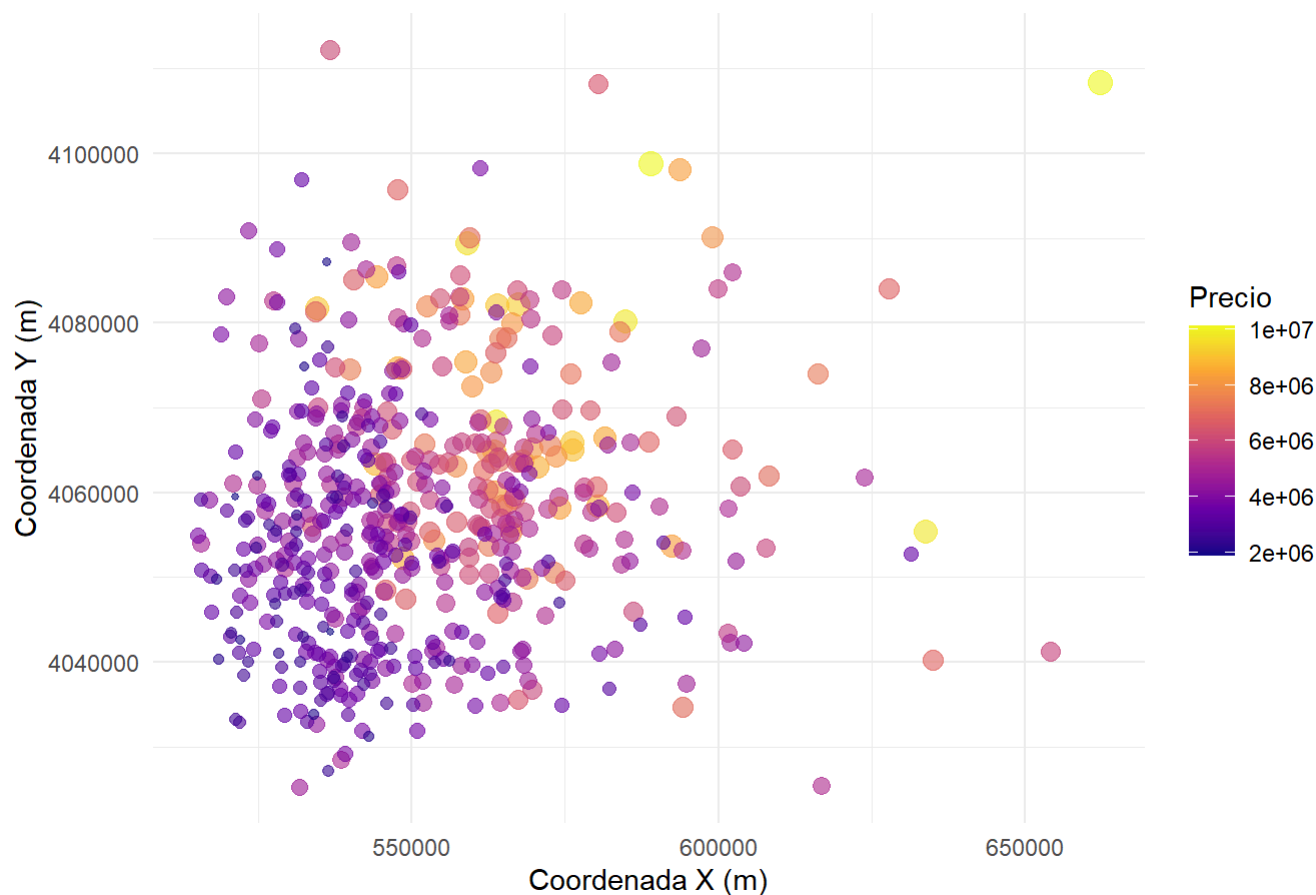
```
## FASE 3: VISUALIZACIÓN ESPACIAL
```

```
cat("-----\n\n")
```

```
## -----
```

```
ggplot(housing_spatial, aes(x = x, y = y)) +
  geom_point(aes(color = price, size = price), alpha = 0.6) +
  scale_color_viridis_c(option = "plasma", name = "Precio") +
  scale_size_continuous(range = c(1, 4), guide = "none") +
  labs(
    title = "Distribución Espacial de Precios",
    x = "Coordenada X (m)",
    y = "Coordenada Y (m)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))
```

Distribución Espacial de Precios



```
# 6. CONVERSIÓN A OBJETO ESPACIAL =====
cat("FASE 4: PREPARACIÓN PARA ANÁLISIS GEOESTADÍSTICO\n")
```

```
## FASE 4: PREPARACIÓN PARA ANÁLISIS GEOESTADÍSTICO
```

```
cat("-----\n\n")
```

```
## -----
```

```
# Convertir a SpatialPointsDataFrame
housing_sp <- housing_spatial
coordinates(housing_sp) <- ~x + y
proj4string(housing_sp) <- CRS("+proj=utm +zone=19 +datum=WGS84")

cat(sprintf("Objeto espacial creado: %d puntos\n", nrow(housing_sp)))
```

```
## Objeto espacial creado: 533 puntos
```

```
cat("Formato: SpatialPointsDataFrame\n\n")
```

```
## Formato: SpatialPointsDataFrame
```

```
# 7. CÁLCULO DEL VARIOGRAMA EMPÍRICO =====
cat("FASE 5: ANÁLISIS DE AUTOCORRELACIÓN ESPACIAL\n")
```

```
## FASE 5: ANÁLISIS DE AUTOCORRELACIÓN ESPACIAL
```

```
cat("-----\n\n")
```

```
## -----
```

```
# Calcular extensión espacial
bbox_data <- bbox(housing_sp)
dist_x <- bbox_data[1,2] - bbox_data[1,1]
dist_y <- bbox_data[2,2] - bbox_data[2,1]
dist_max <- min(dist_x, dist_y) / 2

cat(sprintf("Distancia máxima para variograma: %.0f m\n", dist_max))
```

```
## Distancia máxima para variograma: 43462 m
```

```
# Calcular variograma empírico
vario_empirico <- variogram(
  log_precio ~ 1,
  data = housing_sp,
  cutoff = dist_max,
  width = dist_max / 15
)

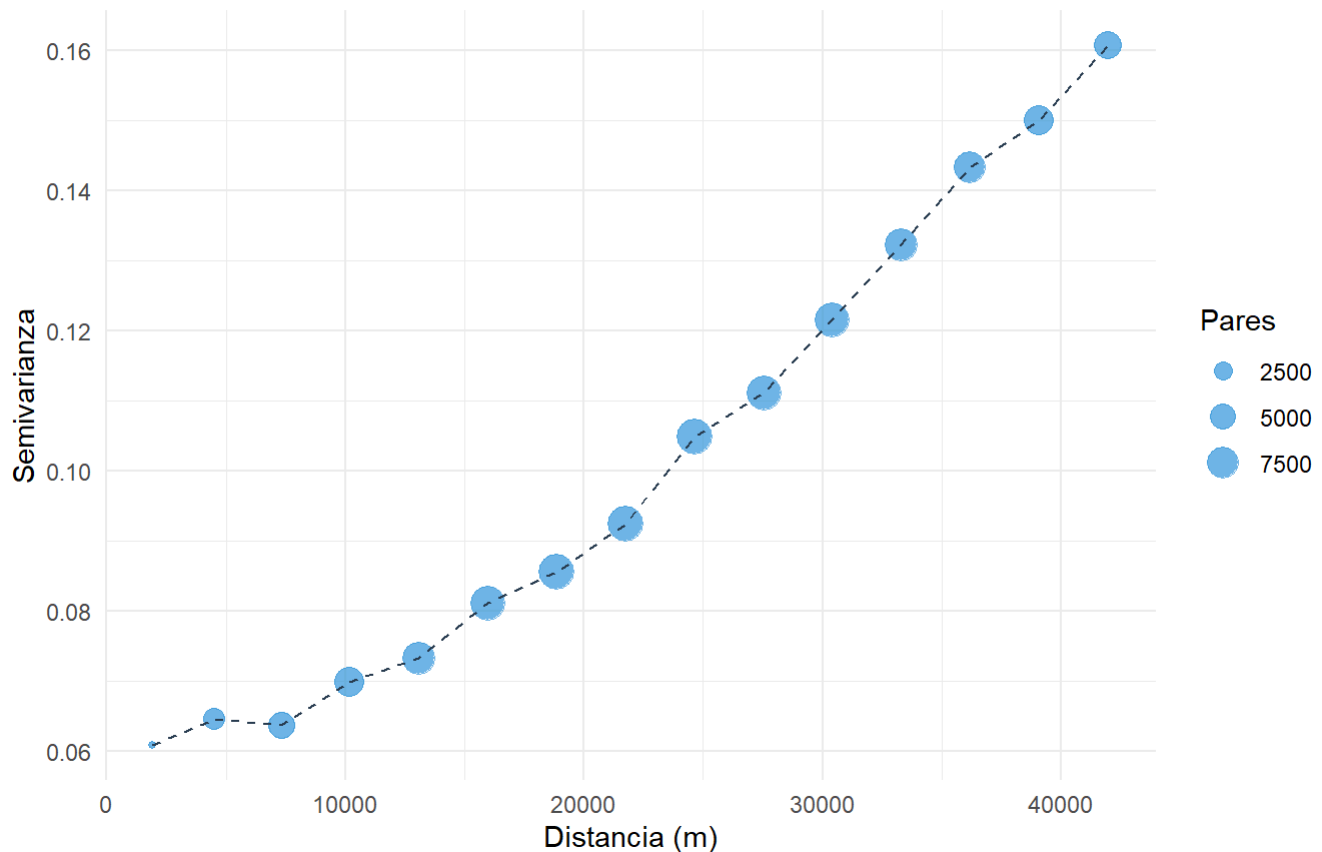
cat(sprintf("Variograma calculado con %d bins\n\n", nrow(vario_empirico)))
```

```
## Variograma calculado con 15 bins
```

```
# Visualizar
ggplot(vario_empirico, aes(x = dist, y = gamma)) +
  geom_point(aes(size = np), color = "#3498db", alpha = 0.7) +
  geom_line(color = "#2c3e50", linetype = "dashed") +
  labs(
    title = "Variograma Empírico",
    subtitle = "Log(Precio de Vivienda)",
    x = "Distancia (m)",
    y = "Semivarianza",
    size = "Pares"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold"))
```

Variograma Empírico

Log(Precio de Vivienda)



```
# 8. AJUSTE DE MODELO DE VARIOGRAMA =====
cat("FASE 6: AJUSTE DE MODELO TEÓRICO\n")
```

```
## FASE 6: AJUSTE DE MODELO TEÓRICO
```

```
cat("-----\n\n")
```

```
## -----
```

```
# Estimar parámetros iniciales del variograma
nugget_init <- min(vario_empirico$gamma) * 0.3
sill_init <- (max(vario_empirico$gamma) - nugget_init) * 0.9
range_init <- dist_max * 0.3

cat("Parámetros iniciales:\n")
```

```
## Parámetros iniciales:
```

```
cat(sprintf(" - Nugget: %.4f\n", nugget_init))
```

```
## - Nugget: 0.0182
```

```
cat(sprintf(" - Sill: %.4f\n", sill_init))
```



```
## - Sill: 0.1282
```

```
cat(sprintf(" - Range: %.0f m\n\n", range_init))
```

```
## - Range: 13039 m
```

```
# Probar modelos
modelos <- c("Exp", "Gau", "Sph")
nombres <- c("Exponencial", "Gaussiano", "Esférico")

cat("Ajustando modelos:\n")
```

```
## Ajustando modelos:
```

```
ajustes <- list()
sse_vals <- c()

for(i in seq_along(modelos)) {
  modelo_nombre <- nombres[i]
  modelo_tipo <- modelos[i]

  cat(sprintf(" %s... ", modelo_nombre))

  tryCatch({
    # Crear modelo inicial
    vgm_init <- vgm(psill = sill_init,
                   model = modelo_tipo,
                   range = range_init,
                   nugget = nugget_init)

    # Ajustar
    vgm_fit <- fit.variogram(vario_empirico, vgm_init)

    # Calcular SSE
    vgm_line <- variogramLine(vgm_fit, maxdist = dist_max)
    pred_gamma <- approx(vgm_line$dist, vgm_line$gamma,
                       xout = vario_empirico$dist)$y
    sse <- sum((vario_empirico$gamma - pred_gamma)^2, na.rm = TRUE)

    ajustes[[modelo_nombre]] <- vgm_fit
    sse_vals[modelo_nombre] <- sse

    cat(sprintf("✓ SSE = %.6f\n", sse))

  }, error = function(e) {
    cat("X Error\n")
  })
}
```

```
## Exponencial...
```

```
## Warning in fit.variogram(vario_empirico, vgm_init): No convergence after 200
## iterations: try different initial values?
```

```
## ✓ SSE = 0.000732
##   Gaussiano... ✓ SSE = 0.000026
##   Esférico...
```

```
## Warning in fit.variogram(vario_empirico, vgm_init): No convergence after 200
## iterations: try different initial values?
```

```
## ✓ SSE = 0.003776
```

```
# Seleccionar mejor modelo
if(length(ajustes) > 0) {
  mejor_idx <- which.min(sse_vals)
  mejor_nombre <- names(sse_vals)[mejor_idx]
  modelo_final <- ajustes[[mejor_nombre]]

  cat(sprintf("\nMejor modelo: %s (SSE = %.6f)\n", mejor_nombre, min(sse_vals)))
} else {
  cat("\nUsando modelo Gaussiano con parámetros iniciales\n")
  modelo_final <- vgm(psill = sill_init, model = "Gau",
                      range = range_init, nugget = nugget_init)
  mejor_nombre <- "Gaussiano"
}
```

```
##
## Mejor modelo: Gaussiano (SSE = 0.000026)
```

```
cat("—————\n")
```

```
## —————
```

```
print(modelo_final)
```

```
##   model      psill    range
## 1   Nug 0.06117418     0.00
## 2   Gau 0.20334924 51219.99
```

```
cat("\n")
```

```
# Extraer parámetros
nugget_final <- ifelse(any(modelo_final$model == "Nug"),
                      modelo_final$psill[modelo_final$model == "Nug"], 0)
params <- modelo_final[modelo_final$model != "Nug",]
sill_parcial <- params$psill[1]
range_final <- params$range[1]
sill_total <- nugget_final + sill_parcial

cat("Parámetros del modelo ajustado:\n")
```

```
## Parámetros del modelo ajustado:
```

```
cat(sprintf(" - Nugget (C0):      %.4f\n", nugget_final))
```

```
## - Nugget (C0):      0.0612
```

```
cat(sprintf(" - Sill parcial (C): %.4f\n", sill_parcial))
```

```
## - Sill parcial (C): 0.2033
```

```
cat(sprintf(" - Sill total:      %.4f\n", sill_total))
```

```
## - Sill total:      0.2645
```

```
cat(sprintf(" - Range:          %.0f m\n", range_final))
```

```
## - Range:          51220 m
```

```
cat(sprintf(" - Dependencia espacial: %.1f%%\n\n",
            (sill_parcial/sill_total)*100))
```

```
## - Dependencia espacial: 76.9%
```

```
# 9. VISUALIZACIÓN DEL AJUSTE =====
cat("FASE 7: VISUALIZACIÓN DEL AJUSTE\n")
```

```
## FASE 7: VISUALIZACIÓN DEL AJUSTE
```

```
cat("-----\n\n")
```

```
## -----
```

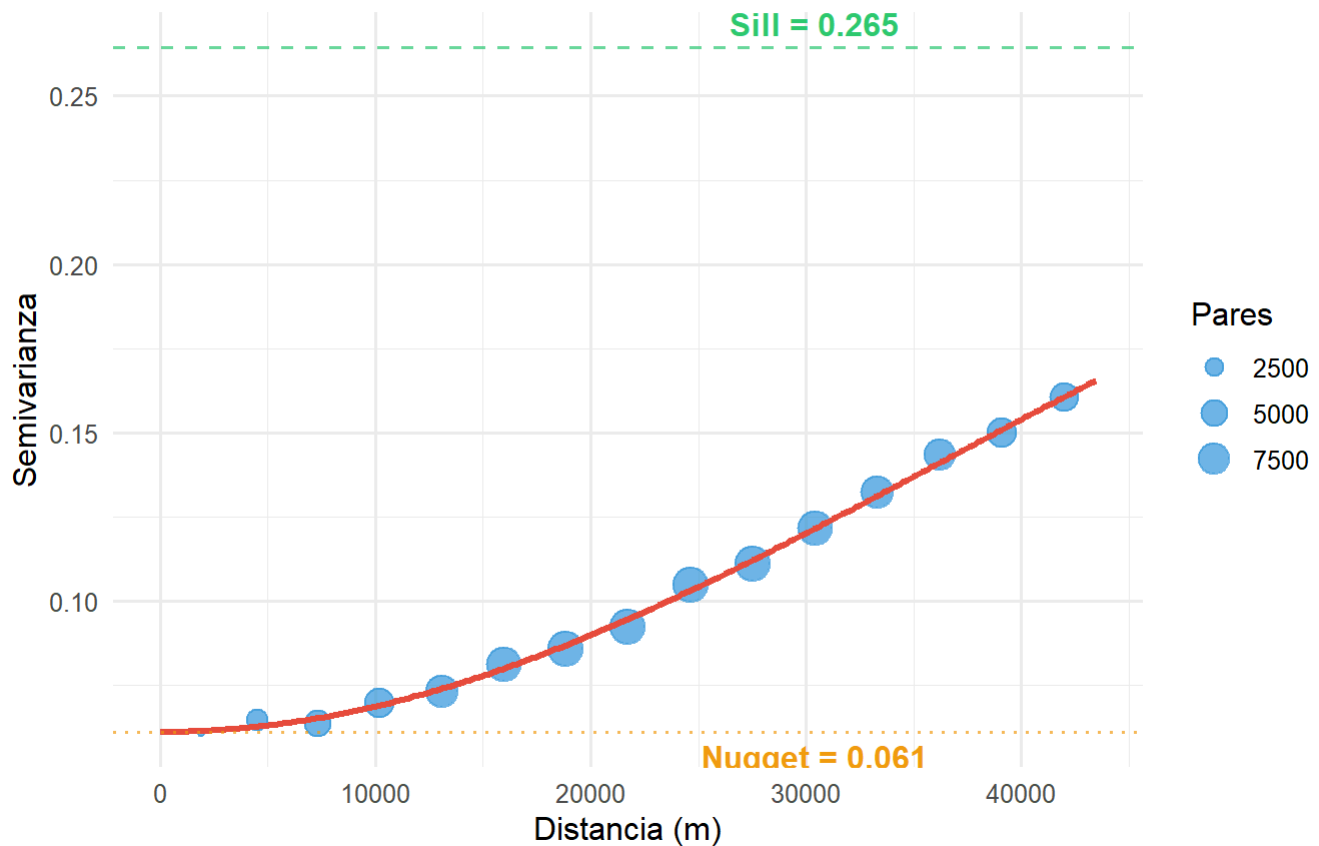
```
# Generar línea del modelo
linea_modelo <- variogramLine(modelo_final, maxdist = dist_max)

ggplot() +
  geom_point(data = vario_empirico,
    aes(x = dist, y = gamma, size = np),
    color = "#3498db", alpha = 0.7) +
  geom_line(data = linea_modelo,
    aes(x = dist, y = gamma),
    color = "#e74c3c", size = 1.2) +
  geom_hline(yintercept = sill_total, linetype = "dashed",
    color = "#2ecc71", alpha = 0.7) +
  geom_hline(yintercept = nugget_final, linetype = "dotted",
    color = "#f39c12", alpha = 0.7) +
  annotate("text", x = dist_max*0.7, y = sill_total,
    label = sprintf("Sill = %.3f", sill_total),
    vjust = -0.5, color = "#2ecc71", fontface = "bold") +
  annotate("text", x = dist_max*0.7, y = nugget_final,
    label = sprintf("Nugget = %.3f", nugget_final),
    vjust = 1.5, color = "#f39c12", fontface = "bold") +
  labs(
    title = paste("Ajuste del Modelo", mejor_nombre),
    subtitle = sprintf("Range = %.0f m", range_final),
    x = "Distancia (m)",
    y = "Semivarianza",
    size = "Pares"
  ) +
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold", size = 14))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Ajuste del Modelo Gaussiano

Range = 51220 m



```
# 10. KRIGING - PREDICCIÓN ESPACIAL =====
cat("FASE 8: KRIGING - PREDICCIÓN ESPACIAL\n")
```

```
## FASE 8: KRIGING - PREDICCIÓN ESPACIAL
```

```
cat("-----\n\n")
```

```
## -----
```

```
# Crear grilla de predicción
n_grid <- 40
grid_x <- seq(bbox_data[1,1], bbox_data[1,2], length.out = n_grid)
grid_y <- seq(bbox_data[2,1], bbox_data[2,2], length.out = n_grid)
grid_df <- expand.grid(x = grid_x, y = grid_y)

# Convertir a objeto espacial
coordinates(grid_df) <- ~x + y
proj4string(grid_df) <- proj4string(housing_sp)
gridded(grid_df) <- TRUE

cat(sprintf("Grilla de predicción: %dx%d = %d puntos\n", n_grid, n_grid, n_grid^2))
```

```
## Grilla de predicción: 40x40 = 1600 puntos
```

```
cat("Ejecutando Kriging Ordinario...\n")
```

```
## Ejecutando Kriging Ordinario...
```

```
# Realizar kriging
kriging_result <- krige(
  log_precio ~ 1,
  locations = housing_sp,
  newdata = grid_df,
  model = modelo_final
)
```

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

```
cat("Kriging completado\n\n")
```

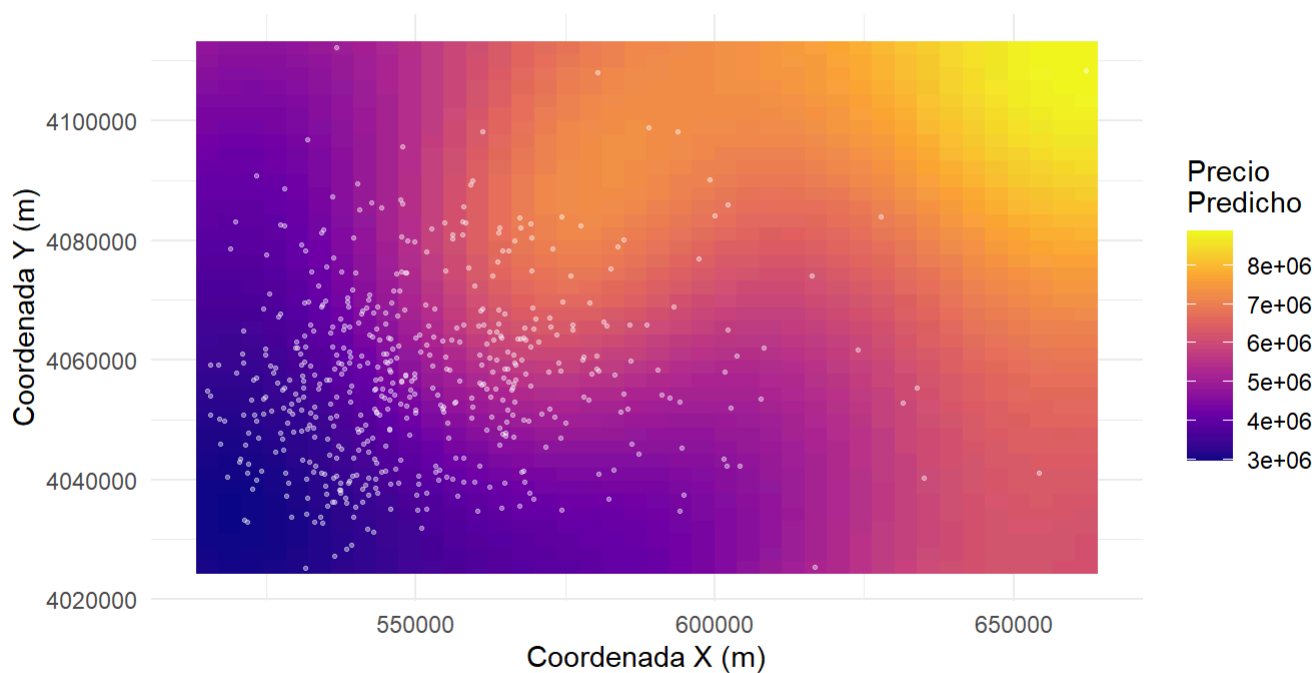
```
## Kriging completado
```

```
# Convertir a data frame
kriging_df <- as.data.frame(kriging_result)
kriging_df$precio_pred <- exp(kriging_df$var1.pred)

# Visualizar predicción
ggplot() +
  geom_tile(data = kriging_df,
            aes(x = x, y = y, fill = precio_pred)) +
  geom_point(data = housing_spatial,
             aes(x = x, y = y),
             size = 0.5, alpha = 0.4, color = "white") +
  scale_fill_viridis_c(option = "plasma", name = "Precio\nPredicho") +
  labs(
    title = "Predicción Espacial - Kriging Ordinario",
    subtitle = paste("Modelo:", mejor_nombre),
    x = "Coordenada X (m)",
    y = "Coordenada Y (m)"
  ) +
  coord_fixed() +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))
```

Predicción Espacial - Kriging Ordinario

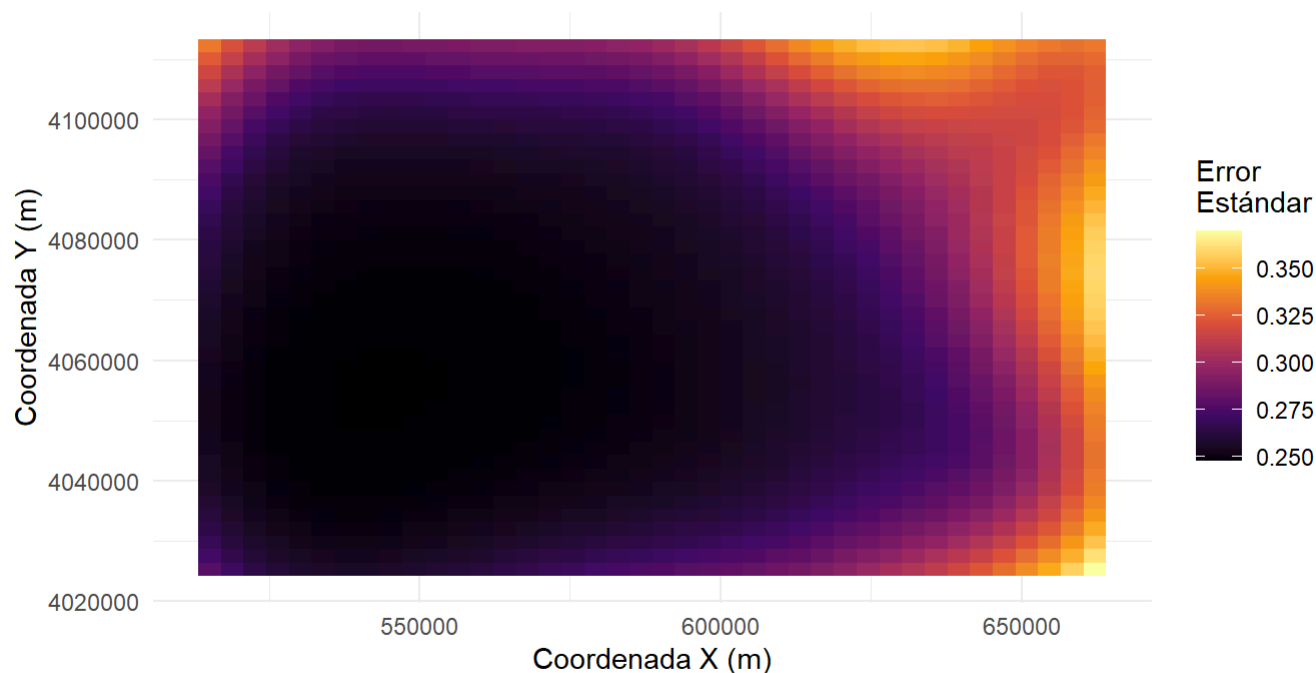
Modelo: Gaussiano



```
# Mapa de incertidumbre
ggplot() +
  geom_tile(data = kriging_df,
            aes(x = x, y = y, fill = sqrt(var1.var))) +
  scale_fill_viridis_c(option = "inferno", name = "Error\nEstándar") +
  labs(
    title = "Mapa de Incertidumbre",
    subtitle = "Error estándar de la predicción",
    x = "Coordenada X (m)",
    y = "Coordenada Y (m)"
  ) +
  coord_fixed() +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", size = 14))
```

Mapa de Incertidumbre

Error estándar de la predicción



```
# 11. RESUMEN FINAL =====
cat("=====\\n")
```

```
## =====
```

```
cat("RESUMEN DEL ANÁLISIS ESPACIAL\\n")
```

```
## RESUMEN DEL ANÁLISIS ESPACIAL
```

```
cat("=====\\n\\n")
```

```
## =====
```

```
cat("Dataset:\\n")
```

```
## Dataset:
```

```
cat(sprintf(" - Observaciones: %d\\n", nrow(housing_spatial)))
```

```
## - Observaciones: 533
```

```
cat(sprintf(" - Variable: Precio de Vivienda\\n\\n"))
```



```
## - Variable: Precio de Vivienda
```

```
cat("Estadísticas:\n")
```

```
## Estadísticas:
```

```
cat(sprintf(" - Media: %.2f\n", mean(housing_spatial$price)))
```

```
## - Media: 4718387.04
```

```
cat(sprintf(" - Desv. Est.: %.2f\n\n", sd(housing_spatial$price)))
```

```
## - Desv. Est.: 1694235.22
```

```
cat("Modelo de variograma:\n")
```

```
## Modelo de variograma:
```

```
cat(sprintf(" - Tipo: %s\n", mejor_nombre))
```

```
## - Tipo: Gaussiano
```

```
cat(sprintf(" - Nugget: %.4f\n", nugget_final))
```

```
## - Nugget: 0.0612
```

```
cat(sprintf(" - Sill: %.4f\n", sill_total))
```

```
## - Sill: 0.2645
```

```
cat(sprintf(" - Range: %.0f m\n", range_final))
```

```
## - Range: 51220 m
```

```
cat(sprintf(" - Dependencia espacial: %.1f%%\n\n",  
            (sill_parcial/sill_total)*100))
```

```
## - Dependencia espacial: 76.9%
```

```
cat("Interpretación:\n")
```

```
## Interpretación:
```

```
if(sill_parcial/sill_total > 0.75) {  
  cat(" → FUERTE dependencia espacial detectada\n")  
} else if(sill_parcial/sill_total > 0.25) {  
  cat(" → MODERADA dependencia espacial detectada\n")  
} else {  
  cat(" → DÉBIL dependencia espacial detectada\n")  
}
```

```
## → FUERTE dependencia espacial detectada
```

```
cat(sprintf(" → Las viviendas presentan correlación espacial hasta %.0f metros\n",  
  range_final))
```

```
## → Las viviendas presentan correlación espacial hasta 51220 metros
```