



FACULTAD DE  
CIENCIAS ECONÓMICAS  
Y DE ADMINISTRACIÓN



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY

## Microsimulación

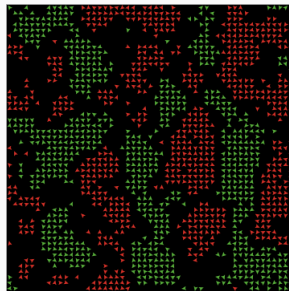
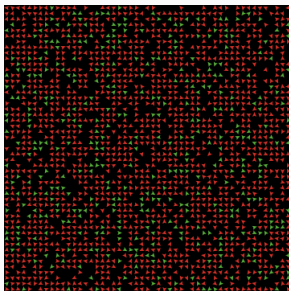
---

Docente: Daniel Ciganda

5<sup>ta</sup> Clase

17 de Septiembre de 2025

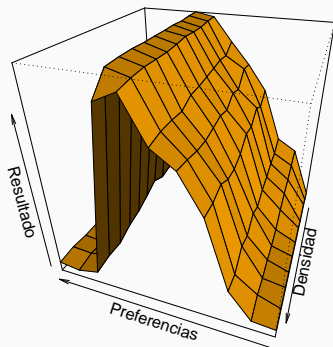
# Modelo de Segregación de Schelling



- Agentes de dos colores diferentes en un mundo-tablero.
- Cada agente tiene un nivel de tolerancia / umbral que determina su felicidad con respecto a la proporción de agentes en su vecindario distintos a si mismo.
- Patrones de segregación **emergen** en el mediano plazo incluso en presencia de agentes relativamente tolerantes.

- Nuestros modelos pueden tener una relación compleja y no obvia entre sus **inputs** (preferencia, densidad) y sus **outputs** (segregación).
- Esta relación forma una "superficie de respuesta" que nos gustaría entender.
- Cada punto en esa superficie requiere ejecutar una simulación que puede ser muy lenta. Explorar miles de combinaciones es computacionalmente **muy costoso o imposible**.

**La pregunta:** ¿Cómo podemos entender esta superficie sin tener que simular cada punto?

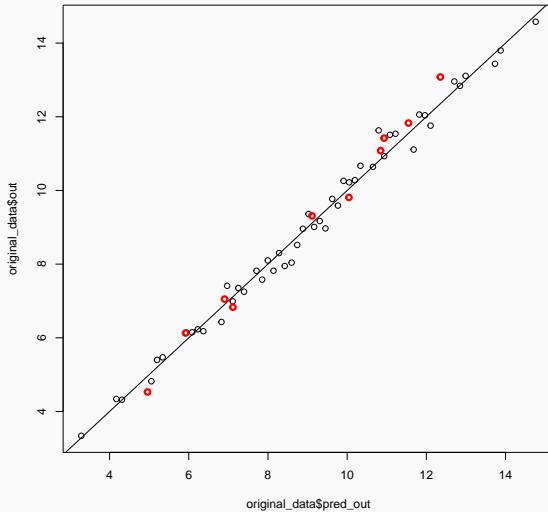


**Figure 1:** Relación inputs-output Modelo de Schelling

- Queremos aproximar una relación desconocida: **outputs** =  $f(\text{inputs})$ , donde **inputs**  $\in \mathbb{R}^p$  y  $p$  puede ser grande.
- **Espacio de parámetros enorme**: las grillas crecen *exponencialmente* con  $p$  (“maldición de la dimensionalidad”).
- Los modelos de simulación suelen ser **costosos** de correr (tiempo/recursos).
- Un **metamodelo** (o emulador) aprende una aproximación de  $f$  para:
  - **explorar** rápido el espacio de parámetros,
  - **mapear** superficies de respuesta,
  - hacer **validación** y **sensibilidad** sin re-simular.

## Cómo construir un Metamodelo (Pasos Generales)

1. **Muestrear el espacio de parámetros inputs:** grilla, muestreo aleatorio, Latin Hypercube, etc.
2. **Ejecutar el simulador** en esos puntos y recolectar **outputs** (ideal: replicar si hay estocasticidad).
3. **Particionar** datos en entrenamiento/validación.
4. **Ajustar varios candidatos** (p.ej., lineal, polinomial, GP, GAM, árboles) sobre el set de entrenamiento.
5. **Evaluar** en validación: predicción vs. observado, residuos, métricas (RMSE,  $R^2$ ).
6. **Seleccionar y diagnosticar** el modelo que mejor balancee sesgo/varianza.
7. **Usar** el metamodelo: mapas de respuesta, sensibilidad, optimización, escenarios.



**Figure 2:** Predicciones del Emulador vs Predicciones del Modelo