

TRABAJO FIN DE MÁSTER:

Una Breve Introducción a la Simulación de Patrones Puntuales Espaciales

JORGE VALERO ZORRAQUINO



**UNIVERSIDAD
DE GRANADA**

FACULTAD DE CIENCIAS

DEPARTAMENTO DE ESTADÍSTICA
E INVESTIGACIÓN OPERATIVA

UNIVERSIDAD DE GRANADA

SEPTIEMBRE 2021

*Memoria realizada por Jorge Valero Zorraquino
en la Facultad de Ciencias de la Universidad de
Granada bajo la dirección de la Doctora María
Dolores Ruiz Medina, Catedrática de la Universi-
dad de Granada, como Trabajo Final del Máster
Universitario en Estadística Aplicada.
Granada, Septiembre de 2021.*

Agradecimientos

En primer lugar a mi tutora, M^a Dolores, por darme la posibilidad de realizar este trabajo y por toda la ayuda recibida.

A todos los profesores del Máster con los que he podido aprender nuevos conocimientos y profundizar en otros tantos.

A todos los compañeros con los que he compartido este curso y que han sido de gran ayuda.

Por supuesto, también a mis padres, hermanos y demás familia y amigos.

Y, especialmente, a mi primo Javier.

¡Muchas gracias a todos!

Introducción

En el siguiente trabajo abordamos una breve revisión de los modelos clásicos de patrones o procesos puntuales espaciales. En particular, adoptaremos como objetivo fundamental la simulación de dichos modelos mediante algoritmos adecuados, que en este trabajo escribiremos en sintaxis R.

Un proceso puntual aleatorio se puede entender como un proceso estocástico cuyas realizaciones consisten en conjuntos de puntos distribuidos aleatoriamente sobre un cierto espacio continuo. Este tipo de procesos son especialmente útiles en el análisis de eventos que ocurren de forma aleatoria en el tiempo, espacio, o bien, en el espacio y tiempo. En particular, el desarrollo de las técnicas estadísticas para el análisis de patrones puntuales espaciales ha tenido un gran auge en el campo de la Geoestadística, la Epidemiología y, en general, en el seno de las Ciencias Medio-Ambientales. Así se observa un extenso desarrollo de la literatura sobre modelización y análisis estadístico de procesos puntuales, destacando, por ejemplo, los textos clásicos [1], [2], [3] y [4].

Según se detalla seguidamente en el esquema de los contenidos del presente Trabajo Fin de Máster (TFM), los procesos de Poisson doblemente estocásticos o procesos de Cox introducidos por [5] que se analizarán posteriormente en este trabajo, han sido ampliamente considerados en la modelización de eventos aleatorios dirigidos por factores medioambientales, aunque no se han aplicado, dada su naturaleza, de forma extensiva a la modelización de esquemas de interacción entre puntos, que determinan la evolución espacial de diferentes fenómenos físicos o naturales (ver, por ejemplo, [3], [6] y [7]).

De hecho, existe una gran actividad investigadora en este área, siendo múltiples los trabajos que plantean extensiones de estos modelos, por ejemplo, en el contexto de procesos marcados, o bien, en el contexto de los procesos puntuales multivariantes. Se destaca asimismo el desarrollo de herramientas estadísticas para la inferencia sobre este tipo de procesos en el espacio y/o tiempo, desde una perspectiva paramétrica y no paramétrica (ver [8], [9], [10], [11], [12], entre otros). También se han adoptado enfoques bayesianos (ver, por ejemplo [13]).

De este modo, el presente trabajo se divide en una parte teórica donde se definen los conceptos necesarios para entender este tipo de procesos y se recopilan algunos de los algoritmos para su generación o simulación y, por otro lado, una parte más práctica donde se desarrolla en código R la implementación de dichos algoritmos.

En el **Capítulo 1** introducimos el marco teórico de los modelos puntuales basándonos principalmente en [3, Capítulos 2,3 y 5] y [14, Capítulo 3]. Definimos y caracterizamos de forma rigurosa este tipo de procesos, estudiando especialmente los llamados procesos de Poisson, que por su importancia merecen una adecuada atención, diferenciando entre homogéneos o no homogéneos según la naturaleza del parámetro intensidad. Además, se introducen también otros procesos de interés como son los procesos de Cox y los procesos Clúster de Poisson

En el **Capítulo 2** comenzamos con un breve recordatorio sobre los métodos básicos de simulación estocástica de distribuciones unidimensionales, incluyendo, en particular, métodos bien conocidos, tales como el método de la transformada inversa, aceptación-rechazo y composición (ver [3], [14, Sección 3.2] y [15, Capítulos 3-4]). A continuación, se exponen diferentes procedi-

mientos para la generación de los procesos de Poisson basados fundamentalmente en la simulación de los tiempos aleatorios de espera entre llegadas, de llegadas, así como en la generación de la distribución condicionada a los valores de la medida puntual aleatoria de recuento.

Adicionalmente, se utilizan o aplican otras técnicas, basadas también en la generación de distribuciones condicionadas, en el espacio, tal es el caso de las técnicas descritas en la generación de procesos de Cox y procesos Clúster de Poisson.

Finalmente, el objetivo del **Capítulo 3** es implementar y aplicar en código R todos las técnicas y métodos descritos en el Capítulo 2. Comenzamos por los algoritmos más básicos de simulación hasta llegar a procedimientos más complejos para la generación de procesos de Poisson no homogéneos o bien, procesos de Cox y procesos Clúster de Poisson. Además, las salidas obtenidas mediante dicha implementación permiten visualizar diferentes patrones puntuales de naturaleza regular o aleatoria, contemplando la atracción entre puntos, así como la repulsión entre los mismos, o bien, la distribución uniforme en el espacio o cualquier recinto bidimensional.

Índice general

| | |
|--|-----------|
| Agradecimientos | III |
| Introducción | V |
| 1. Introducción a los procesos estocásticos puntuales | 1 |
| 1.1. Procesos puntuales | 2 |
| 1.2. Procesos de Poisson | 5 |
| 1.2.1. Proceso de Poisson homogéneo | 5 |
| 1.2.2. Proceso de Poisson no homogéneo | 7 |
| 1.2.3. Transformación de un proceso de Poisson | 7 |
| 1.3. Intensidad | 9 |
| 1.4. Procesos puntuales derivados del proceso de Poisson | 10 |
| 1.4.1. Proceso de Cox | 10 |
| 1.4.2. Proceso Clúster de Poisson | 11 |
| 2. Simulación de procesos puntuales | 13 |
| 2.1. Introducción a los métodos de simulación | 14 |
| 2.2. Principios básicos de la simulación estocástica | 15 |
| 2.2.1. Método de la transformada inversa | 16 |
| 2.2.2. Método de Aceptación-Rechazo | 19 |
| 2.2.3. Método de composición | 21 |
| 2.2.4. Otros métodos para generar distribuciones específicas | 22 |
| 2.3. Simulación de procesos de Poisson | 23 |
| 2.3.1. Procesos de Poisson homogéneos | 23 |
| 2.3.2. Procesos de Poisson no homogéneos | 24 |
| 2.4. Simulación de procesos derivados del proceso de Poisson | 25 |
| 2.4.1. Proceso de Cox | 26 |
| 2.4.2. Proceso Clúster de Poisson | 26 |
| 3. Generación de procesos puntuales mediante R | 27 |
| 3.1. Puntos aleatorios distribuidos uniformemente en una región dada | 27 |
| 3.2. Técnicas básicas de simulación | 29 |
| 3.2.1. Método de la transformada inversa | 30 |
| 3.2.2. Método de Aceptación-Rechazo | 31 |
| 3.2.3. Método de composición | 33 |
| 3.2.4. Otros métodos | 34 |
| 3.3. Procesos de Poisson | 35 |
| 3.3.1. Proceso homogéneo temporal | 35 |
| 3.3.2. Proceso espacial | 38 |

| | |
|--|-----------|
| 3.4. Procesos derivados del proceso de Poisson | 44 |
| 3.4.1. Proceso de Cox | 44 |
| 3.4.2. Proceso Clúster de Poisson | 49 |
| Conclusiones | 55 |
| Bibliografía | 59 |

Capítulo 1

Introducción a los procesos estocásticos puntuales

Un proceso estocástico se define generalmente, dentro de la teoría de la probabilidad, como un conjunto o colección de variables aleatorias. De este modo, la teoría de los procesos estocásticos se focaliza principalmente en el estudio y modelización de sistemas que evolucionan a lo largo del tiempo, o del espacio, de acuerdo a unas leyes no determinísticas, esto es, de carácter aleatorio. Matemáticamente podemos caracterizar un proceso estocástico tal y como se define en [16, Definición 1.19 p. 24], es decir:

Definición 1.1 *Un proceso estocástico es una colección de variables aleatorias $\{X_t(s) : t \in T, s \in S\}$, donde T es un conjunto de índices y S es un espacio muestral común para todas las variables aleatorias. Para un t fijo, $X_t(s)$ denota una única variable aleatoria definida en S . Para un $s \in S$ fijo, $X_t(s)$ corresponde a una función definida en T llamada realización estocástica del proceso.*

Entre los muchos posibles ejemplos de procesos estocásticos se encuentran el número de personas que acuden a consulta médica en un determinado intervalo de tiempo $[0, T]$, el precio de un activo financiero (como puede ser la acción de una empresa) en un día determinado del mes o el número de parados en un mes concreto.

Además, un proceso estocástico puede ser también un conjunto de vectores aleatorios. Por ejemplo, para dos variables aleatorias, la colección de los vectores aleatorios $\{(X_t^1(s), X_t^2(s)) : t \in T, s \in S\}$ es también un proceso estocástico, referido, en este caso, como proceso bidimensional o bivalente.

Cuando se habla de procesos estocásticos, en ocasiones la variable s se omite, de forma que las variables aleatorias se escriben como X_t o $X(t)$ y normalmente el conjunto T hace referencia al tiempo.

Existen diferentes métodos y técnicas para formular y analizar modelos de procesos estocásticos atendiendo a la naturaleza continua o discreta del espacio de parámetros y espacio de estados. Esta distinción determina el tipo de modelo estocástico (discreto o continuo) y las técnicas que pueden ser usadas para estudiar las propiedades del modelo. Entre los tipos de procesos estocásticos más conocidos se incluyen los procesos de Markov de los cuales se derivan otros procesos importantes como son los procesos de Poisson o los procesos de difusión, cada uno de ellos con sus propiedades específicas e idóneos para modelizar diferentes tipos de fenómenos.

Este trabajo se centra fundamentalmente en un tipo de procesos estocásticos concretos: los procesos puntuales. Un proceso puntual aleatorio es un proceso estocástico cuyas realizaciones consisten en conjuntos de puntos distribuidos aleatoriamente sobre un cierto espacio continuo.

Tales puntos suelen corresponder, en la práctica, a los instantes de tiempo en que han ocurrido algunos sucesos de interés, o a las localizaciones en el espacio de ciertos objetos.

En este capítulo se introduce brevemente el marco teórico de los procesos puntuales, que establece las bases para los modelos estadísticos que se han desarrollado posteriormente en la literatura. Fundamentalmente se introducen los modelos puntuales espaciales haciendo especial hincapié en los procesos de Poisson y procesos derivados. Para ello se sigue principalmente el formalismo matemático desarrollado en [14, Capítulo 3] y [3, Capítulos 2,3 y 5].

1.1. Procesos puntuales

Los orígenes de la teoría moderna de los procesos puntuales son algo confusos y difíciles de trazar. Entre la bibliografía revisada destaca [1], donde se realiza un breve estudio retrospectivo de estos inicios, poniendo el foco de atención en aquellas actividades pasadas en las que se pueden apreciar aspectos en común con la teoría moderna. Entre estas actividades se pueden remarcar algunas como:

- Tablas de mortalidad (o de vida), demografía, epidemiología, etc. Este tipo de tablas muestran el número de muertes (o cualquier otra variable) que se producen en un determinado territorio geográfico o localidad.
- Problemas de conteo, relacionados con el recuento del número de eventos que ocurren en un intervalo de tiempo o región, como pueden la ocurrencia de tormentas eléctricas o el número de células sanguíneas en un hemocitómetro.
- El desarrollo de la física de partículas y la necesaria evolución de los detectores de partículas.
- Procesos de transferencia de información en ingeniería de las telecomunicaciones.

Estas son algunas de las líneas de investigación desde las que parece haber surgido toda la estructura matemática moderna de los procesos puntuales y el conjunto de técnicas derivadas. Algunos ejemplos concretos y detallados de la aplicación de este tipo de procesos los podemos encontrar en [3, Capítulo 1]. Así, se explica como Wolpert y Ickstadt (1998) emplearon este tipo de modelos para estudiar la maduración de un bosque de nogales en una parcela cuadrada de 120×120 m en Durham, Carolina del Norte. Mediante procesos puntuales analizaron las posiciones y la evolución de 85 nogales para comprobar como los árboles tendían a aglomerarse formando clústers, probablemente debido a las condiciones ambientales.

Pero antes de todo, conviene explicar de forma rigurosa qué son los procesos puntuales y cómo se caracterizan. La forma más sencilla e intuitiva de introducir los modelos puntuales es refiriendo al caso uniparamétrico, es decir, con índices en el tiempo ([14, Sección 3.1.1]). De esta forma, un proceso puntual representa los sucesivos instantes de tiempo en los que ocurre un evento, por ejemplo las detecciones de un contador Geiger o las llegadas de clientes a una consulta. Un proceso puntual temporal se puede caracterizar de varias formas:

- Mediante los *tiempos de llegada* $T_1 < T_2 < \dots$ en los que ocurre el evento.
- Mediante los *tiempos de espera* $S_1 = T_i - T_{i-1}$ entre llegadas sucesivas.
- Mediante la medida de *recuento* o *conteo* $N_t = \sum_i 1(T_i \leq t)$. En el caso de intervalos $N(a, b] = N_b - N_a$.

En la Figura 1.1 se representa un proceso puntual unidimensional, visualizando dos de las formas de caracterizar el proceso: mediante tiempos de llegada y tiempos de espera.

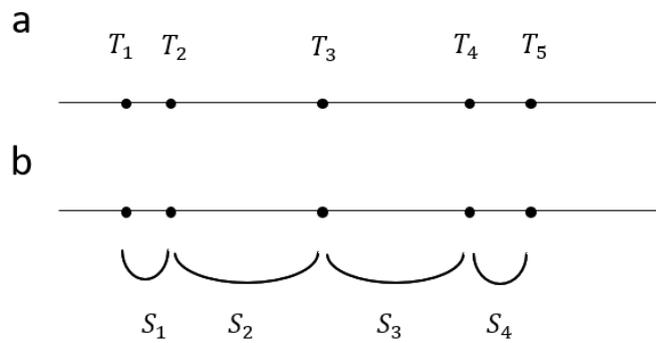


Figura 1.1: Tiempos de llegada (a) y tiempos de espera (b) para un proceso puntual con una única variable temporal.

Tal y como se ha comentado, estas tres propiedades permiten definir un proceso puntual. A modo de ejemplo, podemos hacer uso de ellas para caracterizar un proceso de Poisson homogéneo (que más adelante explicaremos en detalle) con parámetro de intensidad λ en $[0, \infty)$. Este tipo de procesos cumple:

1. Recuentos: $N_t \sim \text{Pois}(\lambda t)$ y $N(a, b] \sim \text{Pois}(\lambda(b - a))$, como consecuencia de esta propiedad se tienen incrementos estacionarios.
2. Incrementos independientes: si los intervalos $(a_1, b_1], \dots, (a_m, b_m]$ son disjuntos, entonces $N(a_1, b_1], \dots, N(a_m, b_m]$ son independientes.
3. Los tiempos aleatorios de espera S_1, S_2, \dots son v.a. i.i.d. (variables aleatorias independientes e idénticamente distribuidas) según una distribución exponencial con media $1/\lambda$.

En la Figura 1.2 se representa la otra forma alternativa de caracterizar un proceso puntual, esto es, mediante el proceso de conteo o recuento. Cada vez que se produce un nuevo evento en el instante de tiempo T_i , el recuento $N(t)$ aumenta en una unidad.

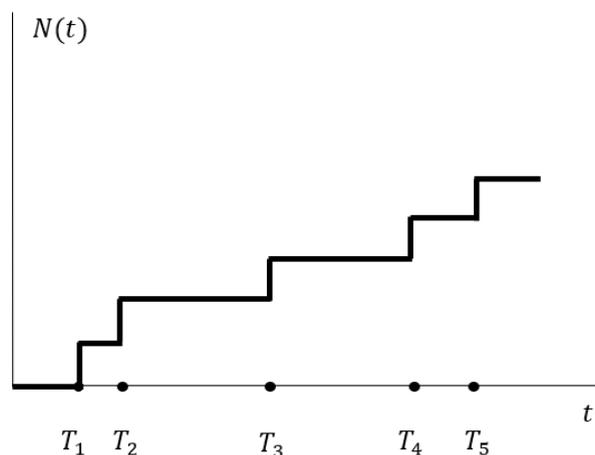


Figura 1.2: Proceso de recuento N_t para un proceso puntual con una única variable temporal. Los T_i son los instantes en los que se produce un nuevo evento.

Esta definición de recuento puede ser generalizada a espacios euclídeos de mayor dimensión. En [3, Sección 2.1] se define un proceso puntual espacial como un subconjunto aleatorio contable de un espacio $S \subseteq \mathbb{R}^d$. Formalmente, en el caso únicamente espacial viene definido como sigue.

Definición 1.2 (Formulación espacial) *Cuando sustituimos el tiempo por el espacio euclídeo \mathbb{R}^d , $d \geq 2$, las cantidades referidas anteriormente como tiempos de espera o tiempos de llegada, ahora expresadas en términos de áreas o distancias respecto a un origen, pueden ser más o menos útiles dependiendo de su interpretación, y por esta razón, se suele utilizar la medida de recuento aleatoria espacial para su caracterización*

$$N(B) = \text{número de puntos que se observan en } B \in \mathbb{R}^d \text{ acotado}$$

Si añadimos además una componente temporal, tenemos la siguiente definición espacio-temporal.

Definición 1.3 (Formulación espacio-temporal) *En el caso de que el proceso tenga tanto una componente espacial como una temporal se puede definir la medida de recuento como*

$$N(B \times [0, T]) = \text{número de puntos aleatorios en } B \in \mathbb{R}^d \text{ acotado en un intervalo de tiempo } [0, T]$$

para $T > 0$.

Finalmente, un proceso puntual se puede definir de forma mucho más general mediante espacios métricos y σ -álgebras ([14, Definición 3.1 p. 51]).

Definición 1.4 (Formulación abstracta) Ψ es un elemento aleatorio de $(\mathcal{N}, \mathcal{A})$, donde \mathcal{N} denota el conjunto de todas las medidas entero-valoradas no negativas μ sobre un espacio métrico completo y separable \mathcal{S} que son finitas sobre cualquier subconjunto compacto $\mathcal{K} \subseteq \mathcal{S}$.

\mathcal{A} denota la más pequeña σ -álgebra sobre \mathcal{N} que contiene $\{\mu : \mu(\mathcal{K}) = n\}$, para cualquier conjunto compacto $\mathcal{K} \subseteq \mathcal{S}$ y cualquier entero $n \geq 0$. $\{\Psi(\mathcal{K}), \mathcal{K} \subseteq \mathcal{S}\}$ es una familia de variables aleatorias.

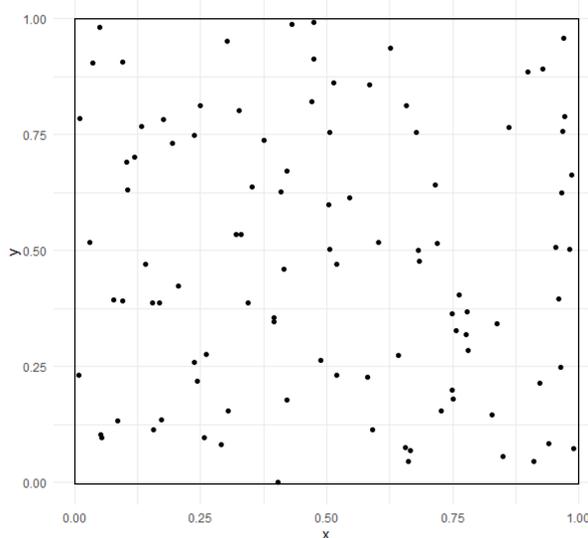


Figura 1.3: Realización de un proceso binomial en una caja unitaria.

A lo largo de este trabajo, nos restringiremos a procesos cuyas realizaciones son localmente finitas, es decir, la medida de recuento es finita para cualquier subconjunto acotado $\mathcal{K} \subseteq \mathcal{S}$. A continuación, se desarrolla un ejemplo clásico de proceso puntual, el proceso binomial, a partir de la formulación abstracta definida arriba.

Ejemplo 1 Sean X_1, \dots, X_n vectores aleatorios con valores en \mathbb{R}^d , i.i.d. uniformemente distribuidas sobre un conjunto acotado $W \subset \mathbb{R}^d$,

$$\Psi(B) = \sum_{i=1}^n \mathbf{1}(X_i \in B)$$

es el número de puntos o sucesos en B . Dada una observación o realización del vector (X_1, \dots, X_n) , $\Psi : B \rightarrow \Psi(B)$ es una medida entero-valuada tal que $\Psi(B) \leq n < \infty$, para cualquier B . Concluimos que Ψ es un elemento de \mathcal{N} .

Para cualquier conjunto espacial compacto $\mathcal{K} \subset \mathbb{R}^d$, $\mathbf{1}(X_i \in \mathcal{K})$ es una variable aleatoria para $i = 1, \dots, n$. Por tanto, $\Psi(\mathcal{K})$ es una variable aleatoria y Ψ define un proceso puntual sobre \mathbb{R}^d . En la Figura 1.3 se muestra un ejemplo de realización de proceso binomial en una caja unitaria de dimensión 2.

Finalmente, introducimos los procesos puntuales simples ([14, Definición 3.2 p. 52]), que son un tipo especial de procesos donde la probabilidad de todos los puntos del proceso de ser distintos es uno. Un proceso puntual simple se puede interpretar también como un conjunto aleatorio cerrado.

Definición 1.5 (Proceso simple) Un proceso puntual Ψ se dice que es simple si

$$\mathbf{P}(\Psi(\{s\}) \leq 1 \quad \forall s \in \mathcal{S}) = 1$$

Entre los procesos puntuales simples se encuentran los procesos binomiales (que anteriormente hemos introducido), de Poisson y de Cox. En este trabajo nos enfocaremos principalmente en este tipo de procesos.

1.2. Procesos de Poisson

Los procesos de Poisson son uno de los procesos puntuales más conocidos y estudiados. Su importancia es enorme, dado su papel esencial en la derivación de resultados teóricos y aplicados, adicionalmente a su componente histórica. En esta sección se detallan las características más importantes de este tipo de procesos así como los modelos más generales y se describen algunas de las transformaciones más interesantes basándonos fundamentalmente en [14, Sección 3.1.2] y [3, Capítulo 3].

1.2.1. Proceso de Poisson homogéneo

En la sección anterior se ha introducido la caracterización de los procesos de Poisson homogéneos como un ejemplo de proceso puntual. A continuación, se presenta una definición más general de dichos procesos ([14, Definición 3.3 p. 53]).

Definición 1.6 El proceso de Poisson homogéneo Π_λ sobre \mathbb{R}^d con intensidad $\lambda > 0$ viene caracterizado por las siguientes propiedades:

1. $\Pi_\lambda(B) \sim \text{Pois}(\nu_d(B)\lambda)$ para cualquier conjunto acotado $B \subset \mathbb{R}^d$

2. $\mathbf{E}[\Pi_\lambda(B)] = \nu_d(B)\lambda$ para cualquier conjunto acotado $B \subset \mathbb{R}^d$

3. $\Pi_\lambda(B_1), \dots, \Pi_\lambda(B_m)$ son independientes cuando B_1, \dots, B_m son conjuntos disjuntos

donde $\nu_d(B)$ es la medida de Lebesgue del conjunto B .

Una característica importante es si el proceso es estacionario o no ([3, Definición 3.4 p. 24]).

Definición 1.7 Un proceso puntual X sobre \mathbb{R}^d es estacionario si su distribución es invariante bajo traslaciones, es decir, la distribución de $X + s = \{x + s : x \in X\}$ es idéntica a la de X para cualquier $s \in \mathbb{R}^d$. Además, se dice que es isotrópico si la distribución es invariante bajo rotaciones sobre el origen de \mathbb{R}^d .

En una dimensión, el proceso de Poisson homogéneo tiene llegadas condicionales uniformes, es decir, dado $N_t = n$ los tiempos de llegada en $[0, t]$

$$T_1 < T_2 < \dots < T_n < t$$

son n variables aleatorias uniformes i.i.d. en $[0, t]$. De forma similar, para dimensiones superiores se cumple la siguiente propiedad ([14, Lema 3.2 p. 53]) que resulta especialmente útil a la hora de generar realizaciones (en los siguientes capítulos se profundizará más en esto).

Lema 1.1 (Propiedad condicional) Dado una observación $\Pi_\lambda(B) = n$, $B \subset \mathbb{R}^d$, $\nu_d(B) < \infty$, la distribución condicional de la restricción de $\Pi_\lambda(B)$ al conjunto B coincide con la distribución del proceso Binomial, definido a partir de n variables aleatorias i.i.d. con distribución uniforme sobre el conjunto $B \subset \mathbb{R}^d$.

En una dimensión, el tiempo de ocurrencia del primer evento en un proceso de Poisson se distribuye de forma exponencial. De forma análoga, para dimensiones superiores se cumple la siguiente propiedad ([14, Lema 3.3 p. 53]).

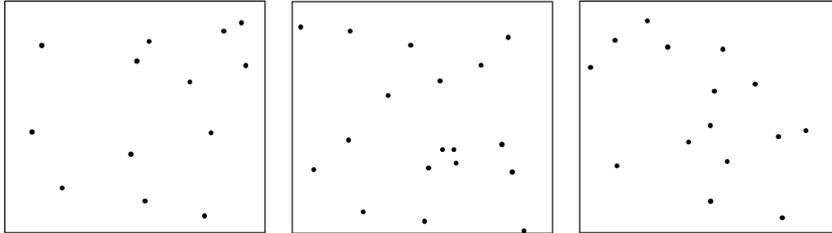


Figura 1.4: Tres realizaciones de un proceso de Poisson homogéneo.

Lema 1.2 (Tiempos de espera exponenciales) La distribución exponencial caracteriza la distancia R aleatoria desde el origen hasta el punto más cercano ubicado en el patrón puntual aleatorio espacial generado por $\Pi_\lambda(B)$ para un proceso de Poisson homogéneo sobre \mathbb{R}^2 .

Para demostrar estos basta observar que $R > r$ si, y solo si, no hay puntos de Π_λ en $B_r(o)$. Por tanto, R tiene función de distribución

$$\begin{aligned} F(r) &= \mathbf{P}(R \leq r) = 1 - \mathbf{P}(R > r) = 1 - \mathbf{P}(N(B_r(o)) = 0) \\ &= 1 - \exp\{-\lambda \nu_2(B_r(o))\} = 1 - \exp\{-\lambda \pi r^2\}. \end{aligned}$$

Esto implica que el área del disco $B_r(o)$ se distribuye exponencialmente.

1.2.2. Proceso de Poisson no homogéneo

Hasta ahora hemos supuesto que la intensidad λ del proceso de Poisson es constante sobre todo el espacio \mathbb{R}^d . Sin embargo, esto se puede generalizar a casos donde la intensidad no es constante, es decir, donde la densidad de puntos varía por el espacio. A continuación, se define un proceso de Poisson no homogéneo ([14, Definición 3.4 p. 54]).

Definición 1.8 *Supongamos que Λ es una medida sobre los elementos de \mathcal{B}^d de \mathbb{R}^d tal que $\Lambda(\mathcal{K}) < \infty$, para cualquier conjunto compacto $\mathcal{K} \subset \mathbb{R}^d$ y $\Lambda(\{x\}) = 0$ para cualquier $x \in \mathbb{R}^d$. Un proceso de Poisson Π espacial no homogéneo con intensidad Λ viene caracterizado por las siguientes propiedades:*

1. $\Pi(B) \sim \text{Pois}(\Lambda(B))$ para cualquier conjunto acotado $B \in \mathcal{B}^d$
2. $\mathbf{E}[\Pi(B)] = \Lambda(B)$ para cualquier conjunto acotado de $B \in \mathcal{B}^d$
3. $\Pi(B_1), \dots, \Pi(B_n)$ son independientes para cualesquiera conjuntos disjuntos $B_1, \dots, B_n \in \mathcal{B}^d$

Esta definición coincide con la del proceso de Poisson homogéneo de densidad $\lambda > 0$ cuando tomamos $\Lambda(\cdot) = \lambda\nu_d(\cdot)$.

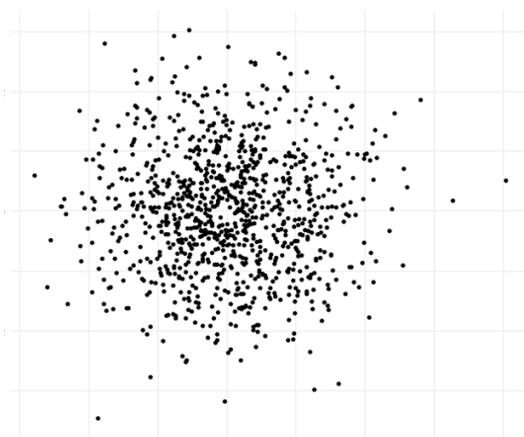


Figura 1.5: Realización de un proceso de Poisson no homogéneo con intensidad proporcional a una gaussiana en el plano.

1.2.3. Transformación de un proceso de Poisson

Supongamos que Π es un proceso de Poisson sobre \mathbb{R}^d con intensidad Λ . Sea $T : \mathbb{R}^d \rightarrow \mathbb{R}^k$ una función continua, entonces bajo ciertas condiciones ([14, Teorema 3.1 p. 55]) se puede definir $(T\Pi)(B) = \Pi(T^{-1}(B))$ como un nuevo proceso espacial de Poisson no homogéneo con intensidad $T\Lambda$ definida como:

$$(T\Lambda)(B) = \Lambda(T^{-1}(B))$$

para todo conjunto acotado $B \subseteq \mathbb{R}^k$.

Teorema 1.1 *Supongamos que Π es un proceso de Poisson sobre \mathbb{R}^d con intensidad Λ . Sea $T : \mathbb{R}^d \rightarrow \mathbb{R}^k$ una función continua tal que $(T\Lambda)(K) = \Lambda(T^{-1}(K)) < \infty$ para todo los conjuntos compactos $K \subset \mathbb{R}^k$ y $(T\Lambda)(\{x\}) = \Lambda(T^{-1}(\{x\})) = 0$ para todo $x \in \mathbb{R}^k$. Entonces la imagen de Π mediante T es un proceso de Poisson sobre \mathbb{R}^k con intensidad $T\Lambda$.*

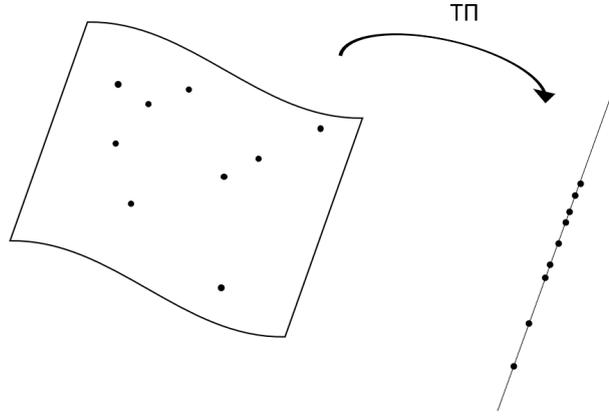


Figura 1.6: Transformación de un proceso de Poisson.

A continuación, se muestra un ejemplo sencillo de transformación desde el caso bidimensional al unidimensional, como puede ser de un proceso espacial a uno temporal.

Ejemplo 2 Sea Π_λ un proceso de Poisson homogéneo en \mathbb{R}^2 con intensidad λ . Sea $T(x) = \|\mathbf{x}\|^2$. Tenemos que $\Lambda(T^{-1}([0, s])) = \Lambda(B_{\sqrt{s}}(o)) = \lambda\pi s < \infty$ para todo $0 < s < \infty$. Por tanto, $T\Pi_\lambda$ es un Proceso de Poisson en $[0, \infty]$ con intensidad $\lambda\pi$. Sea R_k la distancia desde el origen hasta el k -ésimo punto más cercano de Π_λ . Entonces $R_1^2, R_2^2 - R_1^2, R_3^2 - R_2^2, \dots$ son variables aleatorias exponenciales i.i.d con parámetro $\lambda\pi$.

Existen algunas transformaciones que por su utilidad práctica conviene destacar. A continuación, se describen algunas de las más interesantes recogidas en [14, Sección 3.1.2] y [3, Capítulo 2].

Proyección

Consideremos la proyección $T(x_1, x_2) = x_1$ de \mathbb{R}^2 a \mathbb{R} . Sea Π_λ un proceso de Poisson homogéneo en \mathbb{R}^2 con intensidad λ . La proyección de Π_λ no es un proceso puntual porque el número de puntos proyectados en un intervalo compacto $[a, b]$ es infinito:

$$(T\Pi_\lambda)([a, b]) = \Pi_\lambda(T^{-1}([a, b])) = \Pi_\lambda([a, b] \times \mathbb{R}) = \infty$$

En la Figura 1.7 se muestra un ejemplo de proyección de un proceso de Poisson en \mathbb{R}^2 a \mathbb{R} .

Marcado

Consideremos un proceso de Poisson Π_λ en $\mathbb{R}^d \times [0, a]$ con intensidad λ . Puede entenderse como un proceso puntual marcado donde cada punto x_i en \mathbb{R}^d tiene la marca m_i en $[0, a]$. La proyección de Π_λ en \mathbb{R}^d es un proceso de Poisson con intensidad λa ya que en este caso el número de puntos proyectados en cualquier compacto $C \subseteq \mathbb{R}^d$ sí es finito,

$$(T\Pi_\lambda)(C) = \Pi_\lambda(T^{-1}(C)) = \Pi_\lambda(C \times [0, a]) < \infty$$

con intensidad

$$(T\Lambda)(C) = \Lambda(T^{-1}(C)) = \Lambda(C \times [0, a]) = \lambda\nu_d(C \times [0, a]) = \lambda\nu_d(C)a$$

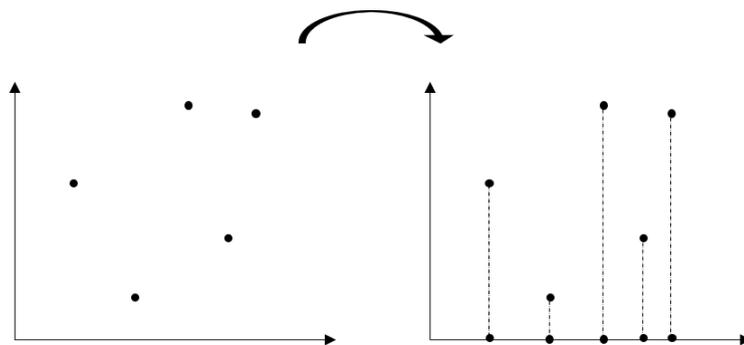


Figura 1.7: Proyección de un proceso de Poisson.

para todo $0 < \nu_d(C) < \infty$. Por lo que $T\Pi_\lambda$ es un proceso de Poisson con intensidad λa .

Esta definición se puede generalizar. Formalmente, dado un espacio M , si una marca aleatoria $m_i \in M$ se le añade a cada punto x_i , entonces

$$X = \{(x_i, m_i) : x_i \in \Pi_\lambda\}$$

es un proceso puntual marcado con puntos en \mathbb{R}^d y marcas en M .

Reducción

Sea Π_λ un proceso de Poisson homogéneo en \mathbb{R}^d con intensidad λ . Supongamos que aleatoriamente eliminamos o mantenemos cada punto de Π_λ , con una probabilidad de eliminación p para cada punto, independientemente de los demás. El proceso Π_λ de los puntos mantenidos es un proceso de Poisson con intensidad $p\lambda$.

Esto se desprende a partir del siguiente lema ([14, Lema 3.1 p. 53]).

Lema 1.3 Si $\eta \sim Pois(\lambda)$ y $(\xi|\eta = n) \sim Binom(n, p)$, entonces $\xi \sim Pois(p\lambda)$, $\eta - \xi \sim Pois((1-p)\lambda)$ y ξ y $\eta - \xi$ son independientes.

1.3. Intensidad

En un proceso puntual unidimensional, la tasa promedio a la que ocurre un suceso aleatorio (puntos) se denomina intensidad. Este parámetro se puede entender como el número esperado de sucesos por unidad de tiempo. En el caso más general, dado un elemento de volumen infinitesimal dx , $\lambda(x)dx$ hace referencia a la probabilidad de que un punto del proceso se encuentre en dicho volumen, ya que, mediante la distribución exponencial se tiene que:

$$P(X \in dx) = 1 - \exp\{-\lambda(x)dx\} \simeq \lambda(x)dx$$

donde se ha aplicado la aproximación de primer orden del desarrollo de Taylor de la exponencial en cero, para valores pequeños del argumento. En el caso de un proceso puntual binomial sobre un conjunto acotado W , el promedio del número de puntos esperados en un conjunto B es:

$$\mathbf{E}\Psi(B) = \mathbf{E} \sum_i \mathbf{1}(X_i \in B) = \sum_i \mathbf{E}\mathbf{1}(X_i \in B)$$

$$= \sum_i \mathbf{P}(X_i \in B) = nP(X_i \in B) = n \frac{\nu_d(B \cap W)}{\nu_d(W)}$$

donde ν_d es la medida de Lebesgue en \mathbb{R}^d . Se aprecia que el resultado obtenido es básicamente la tasa promedio de puntos sobre el conjunto B .

Este concepto de intensidad puede ser definido para procesos puntuales más generales. En esta sección, definimos el parámetro intensidad a partir de [14, Definición 3.5 p. 57]. Esta definición nos permitirá desarrollar modelos estocásticos puntuales más complejos que se derivan del proceso de Poisson.

Definición 1.9 *Sea Ψ un proceso puntual en un espacio métrico completo y separable \mathcal{S} . Supongamos que el número esperado de puntos en cualquier conjunto compacto $K \subset \mathcal{S}$ es finito, es decir, $\mathbf{E}[\Psi(K)] < \infty$. Entonces existe una medida Λ_Ψ sobre \mathcal{S} , llamada intensidad de Ψ , tal que*

$$\Lambda_\Psi(B) = \mathbf{E}[\Psi(B)]$$

para cualquier conjunto acotado $B \subseteq \mathcal{S}$.

1.4. Procesos puntuales derivados del proceso de Poisson

El proceso de Poisson es un modelo ampliamente utilizado para muchos procesos naturales. Además, es realmente sencillo de analizar y simular. Por este motivo, el proceso de Poisson se utiliza como punto de partida para construir nuevos modelos puntuales más complejos. En esta sección, detallamos algunos de los más interesantes recogidos en [14, Sección 3.1.4] y [3, Capítulo 5].

1.4.1. Proceso de Cox

Los procesos de Poisson son normalmente un tipo de modelos demasiado sencillos para aplicarlos a datos reales, sin embargo, debido a su flexibilidad permite construir modelos mucho más complicados. Un proceso de Cox es un proceso de Poisson cuya intensidad es aleatoria. Este tipo de procesos pueden entenderse como un proceso doblemente estocástico, ya que es una generalización de un proceso de Poisson donde la intensidad λ es en sí misma un proceso estocástico. Matemáticamente se definen como sigue ([14, Definición 3.7 p. 59]).

Definición 1.10 *Sea Λ una medida aleatoria localmente finita sobre \mathbb{R}^d . Dada la observación espacial $\Lambda(\cdot) = \lambda(\cdot)$, se asume que la distribución condicionada de Ψ a dicha realización se distribuye según un proceso de Poisson con intensidad λ . Así, se dice que Ψ es un proceso de Cox definido por la medida aleatoria Λ .*

En el caso concreto donde Λ es determinista, Ψ simplemente se trata de un proceso de Poisson con intensidad $\lambda(\cdot) = \Lambda(\cdot)$.

Se dice que un proceso es ergódico si es posible extraer sus características estadísticas propias a partir de la observación del proceso durante un periodo prolongado. En este sentido, un proceso de Cox Ψ es no ergódico (a menos que la distribución de Λ sea degenerada y Ψ sea Poisson).

Para cualquier proceso de Cox, se puede demostrar que

$$\mathbf{E}[\Psi(B)] = \mathbf{E}[\mathbf{E}[\Psi(B)|\Lambda]]$$

$$\mathbf{var}(\Psi(B)) = \mathbf{var}(\mathbf{E}[\Psi(B)|\Lambda]) + \mathbf{E}[\mathbf{var}(\Psi(B)|\Lambda)] = \mathbf{var}(\Lambda(B)) + \mathbf{E}[\Lambda(B)]$$

$$\mathbf{P}[\Psi(B) = 0] = \mathbf{E}[\mathbf{P}(\Psi(B) = 0|\Lambda)] = \mathbf{E}[\exp\{-\Lambda(B)\}]$$

1.4.2. Proceso Clúster de Poisson

Otro tipo de procesos interesantes son los basados en clústering que en esencia se trata de modelos que generan una primera familia de puntos (clústers) llamadas padres, a partir de los cuales se generan nuevos puntos. Se definen como sigue [14, Definición 3.10 p. 60].

Definición 1.11 *Supongamos que podemos definir, para cualquier $x \in \mathbb{R}^d$, la distribución de un proceso puntual ζ_x conteniendo casi seguro un número finito de puntos, es decir, $\mathbf{P}(\zeta_x(\mathbb{R}^d) < \infty) = 1$. Sea Π un proceso de Poisson (homogéneo o no) en \mathbb{R}^d . Dado Π , sea*

$$\Psi = \bigcup_{x_i \in \Pi} \phi_i$$

la superposición de procesos puntuales independientes ϕ_i donde $\phi_i \sim \zeta_{x_i}$. Entonces se dice que Ψ es un proceso Clúster de Poisson con Π como proceso padre y $\{\zeta_x, x \in \mathbb{R}^d\}$ el clústering.

A continuación, se presenta un ejemplo concreto de proceso basado en clústering a partir de un proceso de Poisson.

Proceso Clúster de Matérn

El proceso Clúster de Matérn hace referencia al caso en el que se comienza generando un proceso de Poisson (homogéneo o no) para obtener los padres x_i . A continuación, para cada padre se genera el clúster asociado ζ_{x_i} . Para ello, dado el padre x_i se generan $N_i \sim Pois(\mu)$ puntos aleatorios (proceso de Poisson homogéneo con intensidad μ) i.i.d. y distribuidos uniformemente sobre el disco $B_r(x_i)$ centrado en el padre x_i y de radio r . La característica principal de este proceso es el hecho de que los puntos generados en cada clúster se distribuyen uniformemente en sus respectivos discos.

Capítulo 2

Simulación de procesos puntuales

La simulación de sistemas, fenómenos naturales, fenómenos físicos, etc. proporciona una herramienta muy potente en diferentes áreas científicas. Tal y como se recoge en [17], muchos de los objetos que utilizamos han sido simulados cuidadosamente antes de ser producidos físicamente. Por ejemplo, la gente joven emplea horas haciendo uso de ocio digital (como videojuegos y otras tecnologías) donde se reproducen diferentes deportes o mundos imaginarios, médicos practican con pacientes virtuales, procesos moleculares simulados permiten a las farmacéuticas crear nuevos medicamentos y modelos climáticos predicen de forma cada vez más precisa los cambios meteorológicos.

En el campo de la investigación científica la simulación es una herramienta esencial hoy en día. La ciencia es, por tanto, el ámbito que con más fuerza empuja los límites tecnológicos de las simulaciones hacia el extremo.

De hecho, simulaciones de formación de galaxias, dinámica molecular, plegamiento de proteínas, corrientes oceánicas y diseños aerodinámicos requieren del uso de algoritmos numéricos avanzados y herramientas potentes de procesado de datos. Además, la simulación no solo está transformando la práctica científica, si no que también está llevando a los científicos a revisar y estudiar las relaciones entre modelos, teorías y experimentos.

Esta vasta aplicación de la simulación provoca que cuando uno intenta buscar una definición formal de qué es la simulación, se encuentra con una inmensa colección de conceptos y enfoques dependiendo del ámbito o la perspectiva desde la que se sitúa. Así, resultan llamativos estudios como [18], donde se intenta hacer una recopilación de definiciones de lo que es la simulación dadas por científicos e investigadores que trabajan en el día a día con ellas.

El resumen o la conclusión a la que uno llega tras leer muchas de estas definiciones es que la simulación es, en pocas palabras, la experimentación con modelos. Es decir, el objetivo principal de la simulación es la reproducción de un aspecto concreto de una observación o posible realidad. En su sentido técnico, por tanto, implica la utilización de un modelo para producir resultados, en lugar de experimentar con sistemas reales bajo condiciones determinadas.

Además, si el modelo tiene una componente estocástica, es decir, de aleatoriedad, se trata entonces de una simulación estocástica. Este tipo de simulaciones son de vital importancia ya que son las que se van a revisar, estudiar y recrear a lo largo de todo este trabajo.

En cuanto a las herramientas empleadas en la simulación, se observa que todas estas técnicas permiten reproducir de forma óptima escenarios probabilísticos y estadísticos para poder, a continuación, inferir el comportamiento de un proceso concreto o un sistema aleatorio.

Específicamente, en el ámbito de la probabilidad (ya que es el que nos ocupa fundamentalmente), la generación de modelos de variables aleatorias, procesos estocásticos, campos aleatorios y ecuaciones diferenciales estocásticas ha permitido comprobar la validez de hipótesis sobre el comportamiento de fenómenos y sistemas representados por dichos modelos, así como:

- Explorar características de diferentes distribuciones, ya que en ocasiones los métodos analíticos y numéricos no proporcionan información explícita.
- Inferir comportamientos locales y a gran escala de procesos aleatorios definidos en el tiempo o en el tiempo-espacio.
- Aproximar los estados de un sistema.
- Aproximar el comportamiento de la solución de un modelo diferencial estocástico

Además, en el ámbito estadístico, la simulación estocástica proporciona, entre otros:

- Métodos de generación de muestras con distribuciones o características específicas.
- Métodos para la implementación de técnicas estadísticas que requieren la resolución previa de problemas de optimización.
- Métodos de comprobación de resultados estadísticos asintóticos.
- Métodos de validación de modelos.

En definitiva, la simulación ofrece una alternativa a los métodos analíticos y numéricos, cuando dichos métodos no proporcionan soluciones factibles, o bien, requieren de la aplicación de técnicas o herramientas muy complejas.

En este capítulo se comienza con una revisión bibliográfica de los métodos de simulación estocástica para pasar posteriormente a introducir los principios básicos de la simulación de procesos puntuales. Principalmente se abordan los métodos de simulación para procesos de Poisson y procesos derivados a éste siguiendo principalmente el trabajo de [14, Sección 3.2], [15, Capítulos 3-4] y [3].

2.1. Introducción a los métodos de simulación

En esta sección, se proporciona una breve revisión del trabajo desarrollado en el área de la simulación estocástica, centrándonos en su evolución y sus aplicaciones a diferentes campos.

Los orígenes y evolución de la simulación han sido ampliamente estudiados y documentados. En [19] y [20] se lleva a cabo una revisión profunda de las todas las etapas de la simulación, comenzado en sus posibles orígenes en el siglo 18 con el famoso experimento de la aguja de Buffon, pasando por su explosión debido al gran avance tecnológico resultado de la segunda guerra mundial, hasta llegar a su máximo desarrollo y formalización a partir de los años 70 debido principalmente al progreso en los métodos computacionales y numéricos.

El desarrollo de las técnicas de simulación está inevitablemente ligado al conjunto de actividades o ámbitos donde más beneficios y más uso se le puede dar. De este modo, autores como [21] y [22] analizan también este crecimiento o evolución de las técnicas de simulación pero desde un punto mucho más sectorial, en concreto, desde el punto de vista de la investigación clínica y del desarrollo de los videojuegos respectivamente. Aunque los métodos de simulación empleados en estas áreas se alejan bastante del alcance de este trabajo, es interesante observar como han evolucionado las técnicas de simulación estocástica, cubriendo un gran número de áreas científicas.

En este sentido, [17] desarrolla ampliamente los beneficios del uso de la simulación en el proceso de aprendizaje y en los procesos cognitivos, explicando como la utilización de modelos permite un mejor entendimiento de la realidad y ayuda, de esta forma, a cuestionar y validar la adecuación de los modelos propuestos y, en última instancia, a su mejora y optimización. Otros autores (ver por ejemplo [23]) también analizan el impacto de la simulación en el desarrollo de

los conceptos teóricos y proporcionan una hoja de ruta para el desarrollo de teorías mediante la utilización de métodos de simulación.

Sin embargo, cuando se habla de simulación es imprescindible poner el foco ahí donde verdaderamente ha habido un salto cualitativo en cuanto a prestaciones que ha posibilitado el posterior desarrollo de la simulación, es decir, en las ciencias de la computación. El desarrollo de las técnicas de computación es, sin duda, el gran responsable del crecimiento de los métodos de simulación. La gran capacidad de resolución de cálculos en poco tiempo ha permitido que se lleven a cabo procesos antes imposibles. En [24], [25] y [26] se detallan muchos de los procesos de simulación computacionales que hoy en día son ampliamente utilizados en diferentes ámbitos para resolver problemas relacionados con optimización, integración, resolución de ecuaciones diferenciales, etc.

Otra rama fundamental en el progreso de la simulación (especialmente de la simulación estocástica que es el tema principal de este trabajo) es el desarrollo de técnicas para la generación de número aleatorios. Este tipo de técnicas son la base de la simulación estocástica. En [27] y [28] se realiza una revisión histórica de la evolución de las técnicas de generación de número pseudoaleatorios, describiendo los orígenes. En [29] se desarrollan algunos de los algoritmos actuales para la generación de números pseudoaleatorios y se introduce el método de Monte-Carlo que en ocasiones es considerado como una de las primeras técnicas de simulación estocástica desarrolladas y es utilizado de forma frecuente para gran variedad de problemas matemáticos como la integración o el cálculo de incertidumbres (ver [30]).

Todo este repaso nos lleva finalmente hasta el objeto de interés de este trabajo, es decir, la simulación estocástica. Este tipo de simulaciones tiene una fuerte componente probabilística ya que se fundamenta en el manejo y transformación de variables aleatorias, funciones de distribución de probabilidad, etc. Cabe decir que el formalismo teórico de todas las técnicas estocásticas hace décadas que está bien formalizado y definido, sin embargo, no ha sido hasta el desarrollo avanzado de los métodos computacionales cuando ha sido posible aplicarlas en la práctica. En [31] se desarrolla el marco teórico, definiendo algunos de los procesos estocásticos más habituales como es el proceso de Poisson. Existe una gran cantidad de bibliografía sobre el tema de simulación estocástica. Algunas referencias clásicas pueden encontrarse en [15], [16], [32] o [33]. En ellas se desarrollan los algoritmos clásicos para la generación de variables aleatorias que en esencia se resumen en unos pocos principios básicos que introduciremos en la siguiente sección. Existen otras referencias más centradas en el tipo de procesos que simularemos en este trabajo (procesos puntuales) como [3], [14] o, con una orientación más práctica, [34].

Las aplicaciones de los modelos y simulación estocásticos son absolutamente amplias y abarcan una infinidad de ramas. Entre algunas de ellas podemos encontrarnos con modelos que permiten el análisis de activos financieros (ver [35]), el estudio de la cinética molecular y otros procesos químicos (ver [36], [37]) o el estudio de eventos atmosféricos y geológicos (ver [38], [39], [40]).

2.2. Principios básicos de la simulación estocástica

Como se comentó anteriormente, el objetivo inicial de la simulación estocástica es, en esencia, la generación de situaciones y escenarios asociados a procesos o sistemas con un funcionamiento aleatorio, donde la incertidumbre inherente impide conocer a priori su estado con el fin último de reproducir un sistema real para su posterior análisis. El primer requisito necesario para una simulación estocástica es una fuente de aleatoriedad. Esto se suele dar por sentado pero tiene una importancia fundamental.

Originalmente, fue necesario crear dispositivos computacionales o físicos que permitían generar secuencias de números que no obedecían a ninguna ley o patrón determinístico, es decir, eran de naturaleza aleatoria. Como se comentaba en el Capítulo 1, muchas veces estas secuencias eran

recogidas en tablas (tablas de mortalidad, de vida, etc). Es importante señalar que inicialmente los dispositivos físicos empleados, tales como el lanzamiento de una moneda, la ruleta, etc. sí que proporcionaban secuencias de números realmente aleatorios. Sin embargo, tanto los costes asociados como el tiempo de generación necesario hacían inviable su uso.

Sin embargo, los métodos computacionales actuales ofrecen técnicas y métodos sencillos, de fácil implementación y bajo costo, que ocupan poca memoria en el ordenador, que pueden ser repetidas tantas veces como se desee, etc. Lamentablemente muchas de las llamadas funciones aleatorias proporcionadas por los típicos ordenadores comerciales están lejos de ser aleatorias. Los ordenadores digitales no pueden de forma sencilla imitar un verdadero fenómeno aleatorio como es el ruido electrónico en un diodo. Todas las funciones aleatorias usadas comúnmente son de hecho pseudoaleatorias, lo cual quiere decir que son determinísticas, pero imitan las propiedades de una secuencia de variables aleatorias independientes y uniformemente distribuidas.

Entre los objetivos de este trabajo no se encuentra la revisión y estudio de los métodos usuales de generación de números pseudoaleatorios uniformes, si no más bien la generación de modelos estocásticos para su estudio y aplicación práctica. Por este motivo, debemos comenzar introduciendo y planteando el problema en cuestión.

Supongamos que tenemos un conjunto de variables aleatorias i.i.d. η_1, η_2, \dots que están uniformemente distribuidas en el intervalo $[0, 1]$, es decir, $\eta_i \sim \text{Unif}[0, 1]$. El objetivo es generar una variable aleatoria ξ con una determinada función de probabilidad a partir de las variables η_i .

Las tres técnicas o métodos más usuales para la generación de variables aleatorias son:

- Método de la transformada inversa
- Método de Aceptación-Rechazo
- Método de composición

Conviene señalar que todas estas técnicas son aplicables tanto para el caso discreto como para el continuo. Además, de estos tres métodos, existen otros que aunque usados en menor medida o para casos concretos, también son interesantes.

2.2.1. Método de la transformada inversa

Este método consiste en generar muestra aleatoria U distribuida uniformemente entre 0 y 1, de forma que estos valores se pueden interpretar como probabilidades para, a continuación, devolver el valor x tal que la probabilidad de que la variable aleatoria X (la cual se quiere generar) sea menor que x es $P(X \leq x) = U$.

El método de la transformada inversa para la generación de variables aleatorias continuas es fácilmente implementable, cuando la función de distribución de probabilidad es inyectiva. Si $\eta \sim \text{Unif}[0, 1]$ y tomamos $\xi = a + (b - a)\eta$ donde $a < b$, se desprende claramente que ξ es uniformemente distribuida en $[a, b]$. Formalmente se define como sigue ([14, Definición 3.22 p. 77]).

Definición 2.1 Sea ξ un elemento aleatorio de un espacio \mathcal{X} y $T : \mathcal{X} \rightarrow \mathcal{Y}$ una función medible. Sea $\tau = T(\xi)$. La distribución de ξ viene dada por

$$\mathbf{P}(\tau \in A) = \mathbf{P}(T(\xi) \in A) = \mathbf{P}(\xi \in T^{-1}(A))$$

donde $T^{-1}(A) = \{x \in \mathcal{X} : T(x) \in A\}$ para todos los conjuntos medibles $A \subseteq \mathcal{Y}$.

En el siguiente lema ([14, Lema 3.9 p. 77]) se exponen dos propiedades importantes en simulación. La primera propiedad proporciona el método para la generación de variables aleatorias, mientras que la segunda propiedad se utiliza para comprobar si un conjunto de datos observados o recogidos se ajustan a un modelo específico.

Lema 2.1 *Sea ξ una variable aleatoria con función de distribución acumulada (c.d.f.) $F(x) = \mathbf{P}(\xi \leq x)$. Definimos la función cuantil continua a derechas*

$$F^{-1}(u) = \min\{x : F(x) \geq u\}.$$

Entonces:

1. *Sea η uniformemente distribuida en $[0, 1]$. Entonces $\zeta = F^{-1}(\eta)$ tiene la misma distribución que ξ .*
2. *Si F^{-1} es continua, entonces $\eta = F(\xi)$ está uniformemente distribuida en $[0, 1]$.*

Ejemplos clásicos de aplicación de este procedimiento son para la generación de una variable aleatoria con distribución exponencial o para variables distribuidas como una Poisson.

Ejemplo 3 (Distribución exponencial) *Dada ξ con una función de densidad $f(x) = \lambda e^{-\lambda x}$ para $x > 0$ y 0 para $x \leq 0$, donde $\lambda > 0$ es la intensidad. Entonces su función de distribución de probabilidad viene dada por $F(x) = 1 - e^{-\lambda x}$ para $x > 0$ y 0 para $x \leq 0$. Así, $F^{-1}(u) = -\log(1 - u)/\lambda$. Si $\eta \sim \text{Unif}[0, 1]$ entonces $\zeta = -\log(1 - \eta)/\lambda$ tiene la misma distribución que ξ . Como $1 - \eta \sim \text{Unif}[0, 1]$ también podemos tomar $\zeta = -\log(\eta)/\lambda$.*

Ejemplo 4 (Distribución de Poisson) *Para generar una observación o valor de $n \sim \text{Pois}(\lambda)$, primero generamos $\eta \sim \text{Unif}[0, 1]$ y, a continuación, aplicando el Lema 2.1 tomamos*

$$N = \min \left\{ n : \eta \leq \sum_{k=0}^n e^{-\lambda} \frac{\lambda^k}{k!} \right\}$$

Otra propiedad importante en la simulación es el cambio de variable ([14, Lema 3.10 p 79]).

Lema 2.2 (Cambio de variables) *Sea ξ un elemento aleatorio de \mathbb{R}^d , $d \geq 1$ con función de densidad de probabilidad $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$. Sea $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ una transformación diferenciable tal que, para cualquier $\mathbf{x} \in \mathbb{R}^d$, el jacobiano $\det DT(\mathbf{x})$ es no nulo. Entonces el vector aleatorio $\zeta = T(\xi)$ tiene densidad de probabilidad*

$$g(\mathbf{y}) = \sum_{\mathbf{x} \in T^{-1}(\mathbf{y})} \frac{f(\mathbf{x})}{\det DT(\mathbf{x})}.$$

A continuación, se detallan algunos modelos típicos que son usados de forma habitual en la generación de procesos estocásticos puntuales.

Puntos aleatorios uniformemente distribuidos

Definición 2.2 *Definimos la distribución uniforme sobre un conjunto A de \mathbb{R}^d mediante la siguiente identidad. Para cualquier conjunto medible B de \mathbb{R}^d :*

$$P(\mathbf{X} \in B) = \int_B f(\mathbf{x}) d\mathbf{x} = \frac{\nu_d(A \cap B)}{\nu_d(A)}.$$

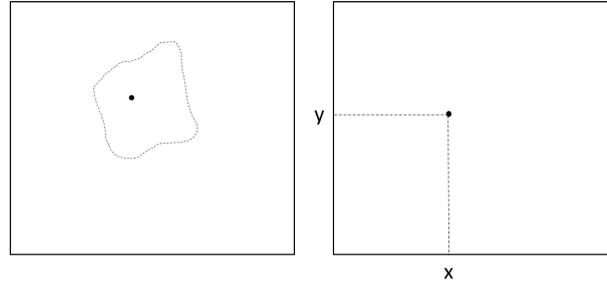


Figura 2.1: (a) Punto aleatorio distribuido uniformemente en un paralelepípedo. (b) Generación de un punto aleatorio en una caja.

Ejemplo 5 (Puntos aleatorios distribuidos uniformemente en una caja) Si X_1, \dots, X_d son variables aleatorias independientes tales que $X_i \sim \text{Unif}[a_i, b_i]$, entonces el vector aleatorio

$$\mathbf{X} = (X_1, \dots, X_d)$$

se distribuye uniformemente sobre el paralelepípedo ([14, Ejemplo 3.18 p. 80])

$$A = \prod_{i=1}^d [a_i, b_i].$$

En la Figura 2.1 se visualiza el proceso de generación de un punto aleatorio uniforme en una caja. Cada una de las componentes x e y se obtiene de forma independiente como variables aleatorias uniformes. Siguiendo la misma idea se pueden generar puntos en una esfera, en este caso haciendo uso de las coordenadas esféricas ([14, Ejemplo 3.19 p. 81]).

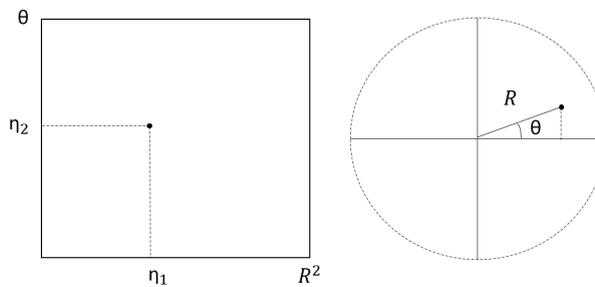


Figura 2.2: Generación de un punto aleatorio uniforme en un disco.

Ejemplo 6 (Punto aleatorio distribuido uniformemente en un disco) Sea \mathbf{X} un punto aleatorio distribuido uniformemente en el disco $B_r(o)$ de radio r centrado en el origen de \mathbb{R}^2 . Consideremos las coordenadas polares

$$R = \sqrt{X_1^2 + X_2^2}, \quad \theta = \arctan(X_2/X_1).$$

R^2 y θ son independientes y uniformemente distribuidas en $[0, r^2]$ y $[0, 2\pi]$ respectivamente. Así, si η_1, η_2 son i.i.d. $\text{Unif}[0, 1]$ y definimos

$$X_1 = r\sqrt{\eta_1} \cos(2\pi\eta_2) \quad X_2 = r\sqrt{\eta_1} \sin(2\pi\eta_2)$$

entonces $\mathbf{X} = (X_1, X_2)$ es un punto aleatorio uniformemente distribuido en $B_r(o)$.

En la Figura 2.2 se visualiza el proceso de generación de un vector aleatorio uniforme en un disco. En este caso se realiza la simulación en coordenadas polares y, a continuación, se transforman los puntos generados a coordenadas cartesianas.

2.2.2. Método de Aceptación-Rechazo

Dentro del método de Aceptación-Rechazo se agrupan una serie de técnicas que en esencia comparten la misma idea aunque en diferentes niveles de complejidad. A continuación, desarrollamos cada una de ellas.

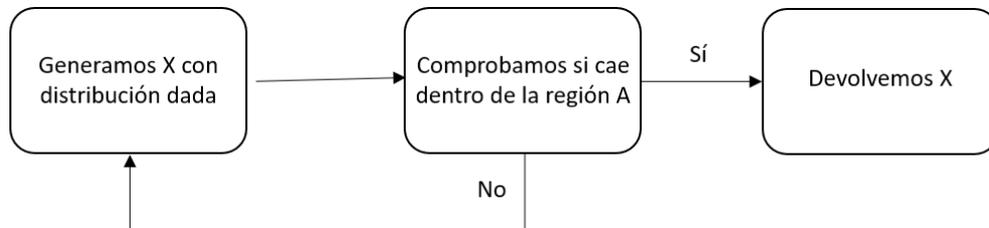


Figura 2.3: Diagrama de flujo del algoritmo de rechazo.

Rechazo

En primer lugar, desarrollamos la técnica de rechazo ([14, Algoritmo 3.2 p. 83]). En la Figura 2.3 se muestra del diagrama de flujo con los pasos requeridos.

Algoritmo 2.1 *Supongamos que queremos generar una realización de una variable aleatoria $\mathbf{X} \in A$, donde A es un subconjunto de los posibles resultados de \mathbf{X} . Asumimos que $\mathbf{P}(\mathbf{X} \in A) > 0$.*

1. *Generamos una realización X de \mathbf{X} .*
2. *Si $X \in A$, se termina el algoritmo y se devuelve X .*
3. *En cualquier otro caso se vuelve al paso 1.*

Ejemplo 7 (Punto aleatorio uniformemente distribuido en una región cualquiera) *Para generar un punto aleatorio \mathbf{X} distribuido uniformemente dentro de una región irrecular $B \subset \mathbb{R}^d$,*

1. *Encerramos B en un conjunto mucho más simple $C \supset B$.*
2. *Usando la técnica de rechazo, generamos puntos aleatorios i.i.d. distribuidos uniformemente dentro de C hasta que caiga en B .*

Este método nos proporciona una forma sencilla de generar puntos en regiones cualesquiera ([14, Lema 3.12 p. 85]).

Lema 2.3 (Distribución condicional en uniformes) *Supongamos que \mathbf{X} está distribuida uniformemente en $A \subset \mathbb{R}^d$ con $\nu_d(A) < \infty$. Sea $B \subset A$. La distribución condicional de \mathbf{X} dado que $\mathbf{X} \in B$ es uniforme en B .*

Sin embargo, en la práctica es un método lento, especialmente cuando se trabaja con regiones complejas y difíciles de parametrizar.

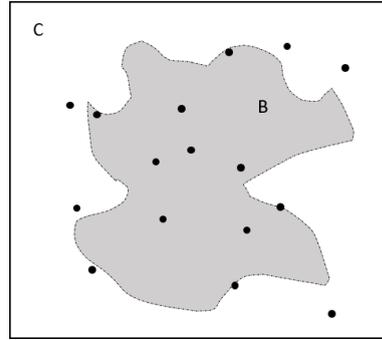


Figura 2.4: Técnica de rechazo para generar puntos aleatorios uniformes en una region irregular B .

Marginalización

En este caso la idea es tomar la función de densidad de la variable de interés para acotar los puntos generados aleatoriamente. Es decir, sea ξ una variable aleatoria real con función de densidad de probabilidad f . Supongamos que $f(x) \leq M$ para todo x . Consideramos el subgrafo

$$A = \{(x, y) : 0 \leq y \leq f(x)\}.$$

Sea (X_1, X_2) un punto aleatorio uniforme en A . La función de densidad conjunta de (X_1, X_2) es $g(x_1, x_2) = \mathbf{1}(0 \leq x_2 \leq f(x_1))$ ya que A tiene área 1. La densidad marginal de X_1 es

$$h(x_1) = \int_0^M \mathbf{1}(x_2 \leq f(x_1)) dx_2 = f(x_1),$$

es decir, X_1 tiene función de densidad f . El algoritmo se presenta a continuación ([14, Algoritmo 3.3 p. 85]).

Algoritmo 2.2 *Supongamos que f es una función de densidad de probabilidad en $[a, b]$ con $\sup_{x \in [a, b]} f(x) < M$.*

1. *Generamos $\xi \sim \text{Unif}[a, b]$.*
2. *Independientemente generamos $\eta \sim \text{Unif}[0, M]$*
3. *Si $\eta < f(\xi)/M$, terminamos y devolvemos el valor ξ . En cualquier otro caso volvemos al paso 1.*

Aceptación-Rechazo

Finalmente, introducimos el método que comúnmente es llamado de Aceptación-Rechazo. Se trata un procedimiento universal que es alternativo al de la transformada inversa y es especialmente útil cuando no se dispone de una expresión sencilla de la función de densidad de la variable aleatoria que se quiere generar, o bien, la función de distribución de probabilidad no es inyectiva.

Queremos generar \mathbf{X} con un función de densidad de probabilidad f compleja. Sea g otra función de densidad que resulta mucho más fácil de simular, tal que $f(\mathbf{x}) \leq Mg(\mathbf{x})$ para todo \mathbf{x} . Este método fue desarrollado por Von Neuman en 1951 ([41]). A continuación, se detalla el algoritmo entero.

Algoritmo 2.3 Sean f y g funciones de densidad y $M < \infty$ tal que $f(\mathbf{x}) \leq Mg(\mathbf{x})$ para todo \mathbf{x} .

1. Generamos \mathbf{Y} con función de densidad g .
2. Independientemente generamos $\eta \sim \text{Unif}[0, M]$.
3. Si $\eta \leq f(\mathbf{Y})/g(\mathbf{Y})$, definimos $\mathbf{X} = \mathbf{Y}$ y terminamos. En cualquier otro caso, volvemos al paso 1.

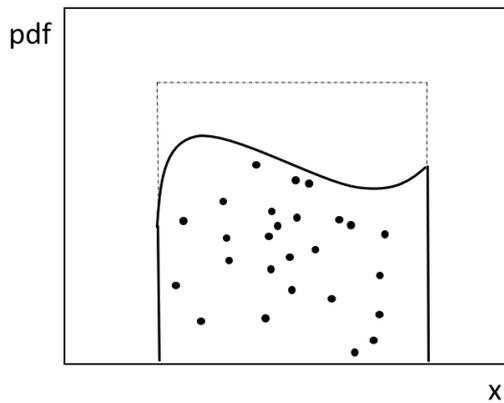


Figura 2.5: Método de Aceptación-Rechazo.

2.2.3. Método de composición

En algunas ocasiones ocurre que la función de densidad de probabilidad de una variable \mathbf{X} de interés se puede escribir como una mixtura de densidades, es decir:

$$f(x) = \sum_{j=1}^k p_j f_j(\mathbf{x})$$

siendo $\sum_{j=1}^k p_j = 1$, $p_j \geq 0$ y f_j funciones de densidad. De este modo se puede generar f simulando en primer lugar p_i y, a continuación, f_j . Este procedimiento es conocido como el método de composición. El algoritmo, que fue también desarrollado por Von Neuman se describe a continuación ([15, Algoritmo 3.7 p. 63]).

Algoritmo 2.4 Podemos generar la variable \mathbf{X} de la siguiente forma:

1. Generamos \mathbf{J} con distribución $\mathbf{P}(J = j) = p_j$ para $j = 1, \dots, k$.
2. Generamos $\mathbf{X} \sim f_j$.

Este algoritmo es muy usado en inferencia bayesiana y para algunas variables de tipo discreto. Un ejemplo clásico para ilustrar este procedimiento es la doble exponencial.

Ejemplo 8 *A partir de la densidad de la distribución doble exponencial:*

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}, \forall x \in \mathbb{R},$$

se deduce que:

$$f(x) = \frac{1}{2}f_1(x) + \frac{1}{2}f_2(x)$$

donde $f_1(x) = \lambda e^{-\lambda x}$ para $x \geq 0$ y cero para $x < 0$, mientras que $f_2(x) = \lambda e^{\lambda x}$ para $x < 0$ y cero para $x \geq 0$. De esta forma, para generar la variable \mathbf{X} basta con generar $\eta_1, \eta_2 \sim Unif(0, 1)$ y si $\eta_1 < 0,5$ devolver $\mathbf{X} = -\ln(1 - \eta_2)/\lambda$, en caso contrario, $\mathbf{X} = \ln(\eta_2)/\lambda$.

2.2.4. Otros métodos para generar distribuciones específicas

Finalmente, presentamos otros métodos de interés para la generación de distribuciones concretas en las que los métodos descritos anteriormente no son los más óptimos basándonos en el trabajo de [15, Capítulo 3].

Box-Muller

El primero de ellos es el método de Box-Muller para la generación de una variable normal. La distribución normal es sin duda una de las más usadas y conocidas. Existe una forma sencilla de simular esta distribución:

$$\mathbf{X} = \left(\sum_{i=1}^{12} U_i - 6 \right)$$

donde $U_i \sim Unif[0, 1]$, \mathbf{X} tiene media cero, varianza uno y se distribuye de forma aproximadamente normal por el teorema central del límite (ver [42]). Aunque la aproximación es aceptable, existe un método exacto para simular la distribución normal, que es conocido como Box-Muller ([15, Algoritmo 3.1 p.54]):

Algoritmo 2.5 *Para generar una variable normal \mathbf{X} :*

1. Generamos $\mathbf{U}_1 \sim Unif[0, 1]$ y tomamos $\theta = 2\pi\mathbf{U}_1$.
2. Generamos $\mathbf{U}_2 \sim Unif[0, 1]$ y tomamos $\xi = -\ln \mathbf{U}_2$, $R = \sqrt{2\xi}$.
3. Tenemos que $\mathbf{X} = R \cos \theta$ y $\mathbf{Y} = R \sin \theta$ son variables aleatorias normales independientes.

Ratio de uniformes

Supongamos que (U, V) es un punto distribuido uniformemente dentro del disco unidad. Se tiene que V/U se distribuye como el ratio de dos variables normales independientes. Además, por el algoritmo de Box-Muller se refiere a la transformación tangente relacionada con la distribución de Cauchy (ver [43]). El algoritmo se detalla a continuación ([15, Algoritmo 3.8 p. 66]).

Algoritmo 2.6 *Para generar una variable de Cauchy \mathbf{X} :*

1. Generamos $\mathbf{U}_1, \mathbf{U}_2 \sim Unif[0, 1]$ y tomamos $V = 2\mathbf{U}_2 - 1$.
2. Si $\mathbf{U}_1^2 + V^2 < 1$ devolvemos $\mathbf{X} = V/\mathbf{U}_1$, en caso contrario, volvemos al paso 1.

2.3. Simulación de procesos de Poisson

En esta sección, se desarrollan diferentes tipos de algoritmos para la simulación de procesos de Poisson. Cada uno de ellos presenta un enfoque distinto, haciendo uso de diferentes propiedades o generalizando a dimensiones superiores. Los métodos descritos se basan en los trabajos de [14, Sección 3.2.2] y, en menor medida, [3].

2.3.1. Procesos de Poisson homogéneos

Comenzamos simulando los procesos de Poisson más sencillos, es decir, la clase de procesos de Poisson homogéneos. A continuación, se presentan cuatro algoritmos diferentes, según usen los tiempos de espera o la generación condicional y sobre diferentes espacios. Estos algoritmos corresponden a [14, Algoritmo 3.5 p. 67], [14, Algoritmo 3.6 p. 67], [14, Algoritmo 3.7 p. 67] y [3, Sección 3.2.3].

Mediante tiempos de espera

Algoritmo 2.7 *Para simular una realización de un proceso de Poisson homogéneo en $[0, a]$:*

1. *Generamos los tiempos de espera entre sucesos S_1, S_2, \dots como variables aleatorias i.i.d. distribuidas como una exponencial de parámetro λ .*
2. *Obtenemos los tiempos de llegada $T_n = \sum_{i=1}^n S_i$.*
3. *Devolvemos el conjunto de tiempos de llegada T_n que verifican $T_n < a$.*

Mediante generación condicional

El proceso de Poisson homogéneo en $[0, \infty]$ tiene llegadas condicionales uniformes, es decir, dado $N_t = n$, los tiempos de llegada en $[0, t]$

$$T_1 < T_2 < \dots < T_n < t$$

son variables aleatorias uniformes i.i.d. en $[0, t]$.

Algoritmo 2.8 *Para simular una realización de un proceso de Poisson homogéneo en $[0, a]$:*

1. *Generamos $N \sim \text{Pois}(\lambda a)$*
2. *Dado $N = n$, generamos n variables i.i.d. $\eta_1, \eta_2, \dots, \eta_n \sim \text{Unif}[0, a]$.*
3. *Ordenamos $\{\eta_1, \dots, \eta_n\}$ para obtener los tiempos de llegada.*

En \mathbb{R}^d mediante generación condicional

Para un proceso de Poisson en \mathbb{R}^d , dado $\Pi_\lambda = n$, la restricción de Π_λ a B tiene la misma distribución que un proceso binomial (n puntos aleatorios i.i.d. distribuidos uniformemente en B).

Estas propiedades pueden usarse directamente para simular un proceso de Poisson en \mathbb{R}^d con intensidad $\lambda > 0$.

Algoritmo 2.9 *Para simular una realización de un proceso de Poisson homogéneo en \mathbb{R}^d :*

1. *Dividimos \mathbb{R}^d en hipercubos unitarios $Q_k, k = 1, 2, \dots$*

2. Generamos variables aleatorias i.i.d. $N_k \sim \text{Pois}(\lambda)$

3. Dado $N_k = n_k$, generamos n_k puntos aleatorios i.i.d. uniformemente distribuidos en Q_k .

Para generar una realización de un Proceso de Poisson Π_λ con intensidad constante λ dentro de una region irregular $B \subset \mathbb{R}^d$, es más sencillo generar un proceso de Poisson Ψ con intensidad λ en un conjunto $C \supset B$ mucho más simple y, a continuación, tomar $\Pi_\lambda = \Psi \cap B$, es decir, quedarse solo con los puntos de Ψ que caen dentro de B .

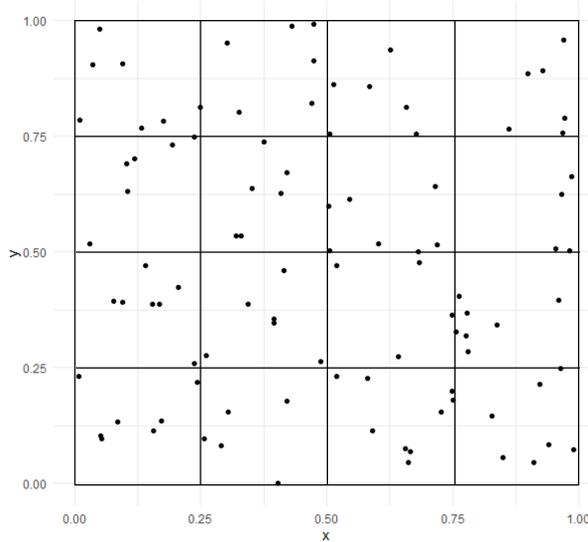


Figura 2.6: Simulación directa de un proceso de Poisson en dos dimensiones.

En la Figura 2.6 se muestra un ejemplo de simulación de un proceso de Poisson en \mathbb{R}^2 donde cada cuadrado hace referencia a un hipercubo Q_k .

2.3.2. Procesos de Poisson no homogéneos

En el caso de procesos de Poisson no homogéneos el procedimiento de simulación se complica notablemente dependiendo de la complejidad de la función λ . En cualquier caso, el número de puntos generados se puede obtener como una variable aleatoria de Poisson. Sin embargo, a la hora de determinar las posiciones de dichos puntos es donde radica la dificultad.

Un método para simular Procesos de Poisson no homogéneos consiste en comenzar simulando uno que sí es homogéneo y, a continuación, transformar adecuadamente los puntos a través de una función determinista. Para ello conviene primero explicar en detalle este tipo de transformaciones ([14, Lema 3.13 p. 88]).

Transformación de procesos de Poisson

Lema 2.4 Sea Π un proceso puntual de Poisson en \mathbb{R}^d , $d \geq 1$ con intensidad $\lambda(u)$, $u \in \mathbb{R}^d$. Sea $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ una transformación diferenciable con $|\det DT(\mathbf{x})| \neq 0$. Entonces $\Psi(\cdot) = (T\Pi)(\cdot)$ es un proceso de Poisson con intensidad

$$\nu(\mathbf{y}) = \sum_{\mathbf{x} \in T^{-1}(\mathbf{y})} \frac{\lambda(\mathbf{x})}{|\det DT(\mathbf{x})|}$$

donde $|\det DT|$ representa el Jacobiano de T . Se deriva aplicando el teorema de cambio de variable introducido anteriormente.

Este lema nos proporciona el procedimiento para transformar procesos de Poisson y determinar sus parámetros de intensidad. A modo de ejemplo, rescatamos el ejemplo visto en la Sección 2.2.1 para aplicar el lema anterior.

Ejemplo 9 (Coordenadas polares) Sea Π_λ un proceso de Poisson homogéneo en \mathbb{R}^2 con intensidad λ . La transformación

$$T(x_1, x_2) = (x_1^2 + x_2^2, \arctan(x_2/x_1))$$

tiene Jacobiano 2. Así, $\Psi = T(\Pi_\lambda)$ es un proceso de Poisson homogéneo en $(0, \infty) \times [0, 2\pi)$ con intensidad $\lambda/2$. Proyectando en la primera componente obtenemos un proceso de Poisson homogéneo con intensidad $\lambda/2 \times 2\pi = \lambda\pi$.

Estas transformaciones permiten construir procesos de Poisson no homogéneos a partir de procesos homogéneos. Sin embargo, en ocasiones el parámetro intensidad λ presenta argumentos no independientes, o equivalentemente, no es factorizable, lo que hace más difícil esta transformación, ya que necesitamos simular cada una de sus componentes teniendo en cuenta la relación entre ellas.

Proceso de Poisson en $B_r^d(o)$ mediante tiempos de espera

Quine y Watson propusieron un método en 1984 ([44]) que es conocido como el método de la simulación radial. Este método se utiliza para generar un proceso de Poisson en la bola $B_r^d(o)$ de radio r , dimensión d y centrada en el origen a partir de los tiempos de espera. En el caso $d = 1$, la bola es un intervalo y el procedimiento es similar al primer algoritmo descrito anteriormente.

Algoritmo 2.10 Para simular una realización de un proceso de Poisson en $B_r^d(o)$:

1. Generamos los tiempos de espera entre sucesos S_1, S_2, \dots como variables aleatorias i.i.d. distribuidas como una exponencial de parámetro $\lambda\nu_d(B_r^d(o))$, donde $\nu_d(B_r^d(o))$ es la medida de Lebesgue de la bola.
2. Generamos variables i.i.d. $\mathbf{U}_1, \mathbf{U}_2, \dots \sim \text{Unif}(B^d(o))$.
3. Tomamos $R_0 = 0$ y $R_i^d = R_{i-1}^d + S_i$, con $i = 1, 2, \dots$ hasta algún m tal que $R_m > r$.
4. Devolvemos los puntos $R_1\mathbf{U}_1, R_2\mathbf{U}_2, \dots$

2.4. Simulación de procesos derivados del proceso de Poisson

Los procesos puntuales derivados del proceso de Poisson, que fueron descritos en el capítulo anterior, son relativamente sencillos de simular. En esta sección proporcionamos los algoritmos de los procesos de Cox y proceso de Clúster de Poisson basándonos en [14, Algoritmo 3.10 p. 91] y [14, Algoritmo 3.11 p. 91].

2.4.1. Proceso de Cox

Algoritmo 2.11 *Para simular un proceso de Cox:*

1. *Generamos una realización de variable aleatoria intensidad Λ .*
2. *Dado Λ , generamos Ψ teniendo en cuenta de que se trata de un proceso de Poisson con intensidad Λ .*

2.4.2. Proceso Clúster de Poisson

Sea Ψ un proceso Clúster de Poisson estacionario formado por un proceso padre de Poisson homogéneo Π_κ de intensidad κ y donde los clústeres se pueden entender como un mecanismo invariante ante traslaciones del tipo $\zeta_x \equiv \zeta_o + x$, donde x es el punto padre y ζ_o es un proceso puntual, finito casi seguramente en \mathbb{R}^d . Para simular una realización de Ψ dentro de una ventana acotada W , es necesario considerar los padres $x_i \in \Pi_\kappa$ que caen tanto fuera de W como dentro.

Algoritmo 2.12 *Si $\zeta_o \subset B_r(o)$ casi seguramente donde r está fijado, entonces*

1. *Generamos una realización x del proceso de Poisson homogéneo con intensidad κ en $W \oplus B_r(o)$.*
2. *Para cada $x_i \in x$ generamos una realización de ζ_{x_i} .*
3. *Tomamos*

$$y = W \cap \bigcup_i \zeta_{x_i}.$$

En la Figura 2.7 se representa el procedimiento para generar este tipo de procesos.

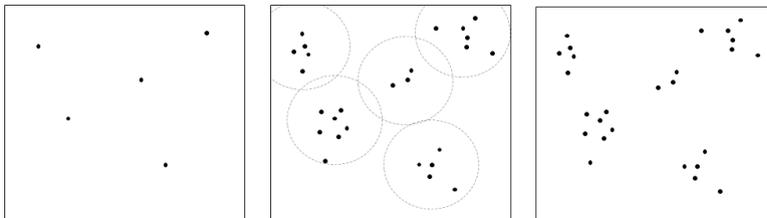


Figura 2.7: Construcción de un proceso Clúster. (a) Proceso de Poisson para generar los puntos padres. (b) Cada padre da lugar a un clúster de puntos. (c) Hasta obtener el conjunto de puntos final.

Capítulo 3

Generación de procesos puntuales mediante R

En este capítulo desarrollamos y aplicamos las técnicas y métodos de simulación descritos en el Capítulo 2 mediante el entorno de programación R. Para ello se proporciona tanto el código en R como los pasos fundamentales que se emplean en cada simulación, además de las representaciones gráficas obtenidas. Algunos de estos algoritmos fueron abordados en la asignatura de Simulación de procesos estocásticos e inferencia estadística del Máster de Estadística Aplicada de la UGR.

Antes de enfocarse en los métodos más importantes y de mayor complejidad, conviene comenzar mostrando y revisando algunos ejemplos de algoritmos sencillos para la generación de procesos puntuales que nos permitirán más adelante simular procesos más complicados como los de Poisson.

3.1. Puntos aleatorios distribuidos uniformemente en una región dada

De esta forma, se presentan a continuación varios ejemplos de generación de puntos aleatorios distribuidos uniformemente sobre diferentes regiones de interés.

Puntos aleatorios uniformemente distribuidos en una caja

Un problema básico es la generación de puntos uniformemente distribuidos dentro de una caja (o hipercubo si se quiere generalizar) de dimensiones conocidas. Para ello basta con generar d (donde d es la dimensión de la caja o hipercubo) variables aleatorias independientes y uniformes $X_i \sim Unif[a_i, b_i]$ (siendo a_i y b_i límites de la caja) y tomar $X = (X_1, \dots, X_d)$ como un punto aleatorio. El algoritmo queda como sigue ([14, Ejemplo 3.18 p. 80]):

1. Generamos $X_i \sim Unif[a_i, b_i]$ variables aleatorias independientes mediante la función de R `runif(d, a, b)`, donde d es el número de variables generadas y a y b los límites para generarlos.
2. Tomamos el punto aleatorio $X = (X_1, \dots, X_d)$.
3. Repetimos hasta obtener tantos puntos aleatorios como se requiera.

A continuación, se muestra el código empleado en R, detallando los pasos y representando gráficamente los resultados:

```

# Punto aleatorio distribuido uniformemente en la caja de lados [0,1] y [0,1]
n <- 100 # Número de puntos generados
# Definimos los límites de la caja
a <- c(0,1)
b <- c(0,1)
u <- runif(n, min = a[1],max = a[2]) # Generamos v.a. uni. como coordenada eje x
v <- runif(n, min = b[1],max = b[2]) # Generamos v.a. uni. como coordenada eje y

df <- data.frame(x =u,y=v) # Creamos dataframe con los puntos

# Representamos gráficamente
library(ggplot2)
ggplot(df, aes(x=x,y=y)) + geom_point() + theme_minimal()

```

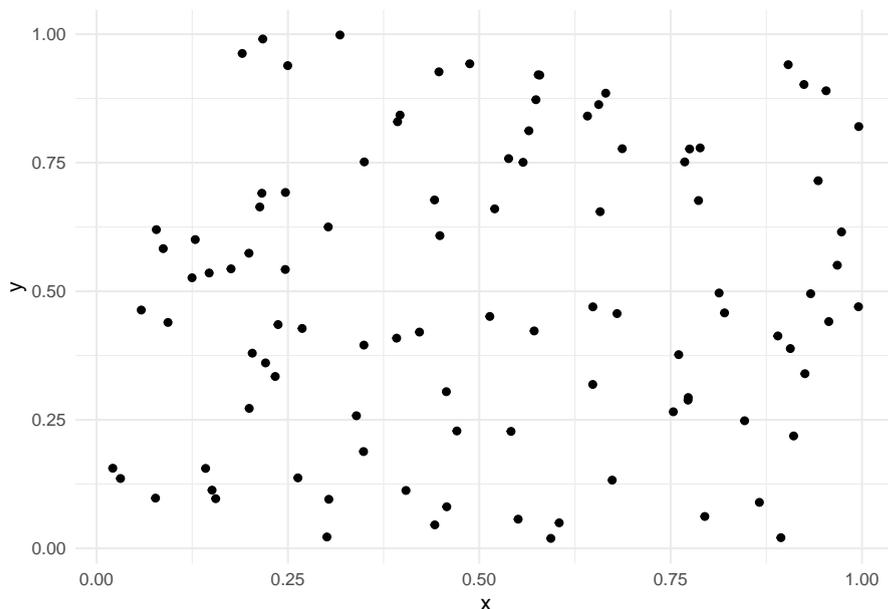


Figura 3.1: Simulación de 100 puntos aleatorios distribuidos uniformemente sobre la caja $[0, 1] \times [0, 1]$.

Puntos aleatorios uniformemente distribuidos en un disco de radio R

El problema anterior se puede generalizar a otro tipo de regiones más complejas, aunque el procedimiento es similar. En este caso, vamos a generar puntos uniformemente distribuidos en un disco de radio R . Para ello procedemos de forma similar al ejemplo anterior ([14, Ejemplo 3.19 p. 81]):

1. Generamos $\eta_1, \eta_2 \sim Unif[0, 1]$ variables aleatorias independientes de forma idéntica al ejemplo anterior.
2. Tomamos el punto aleatorio $X = (R\sqrt{\eta_1} \cos(2\pi\eta_2), R\sqrt{\eta_1} \sin(2\pi\eta_2))$. Esta transformación permite mapear los puntos de la caja al disco de radio R .

3. Repetimos hasta obtener tantos puntos aleatorios como se requiera.

```
# Punto aleatorio distribuido uniformemente en un disco
n <- 1000 # Número de puntos generados
r <- 1 #Definimos el radio del disco

u <- runif(n) #Variable aleatoria uniforme
v <- runif(n) #Variable aleatoria uniforme

# Aplicamos la transformación a polares
x <- r * sqrt(u) * cos(2 * pi * v)
y <- r * sqrt(u) * sin(2 * pi * v)

df <- data.frame(x = x,y = y) # Creamos dataframe con los puntos

# Representamos gráficamente
ggplot(df, aes(x=x,y=y)) + geom_point() + theme_minimal()
```

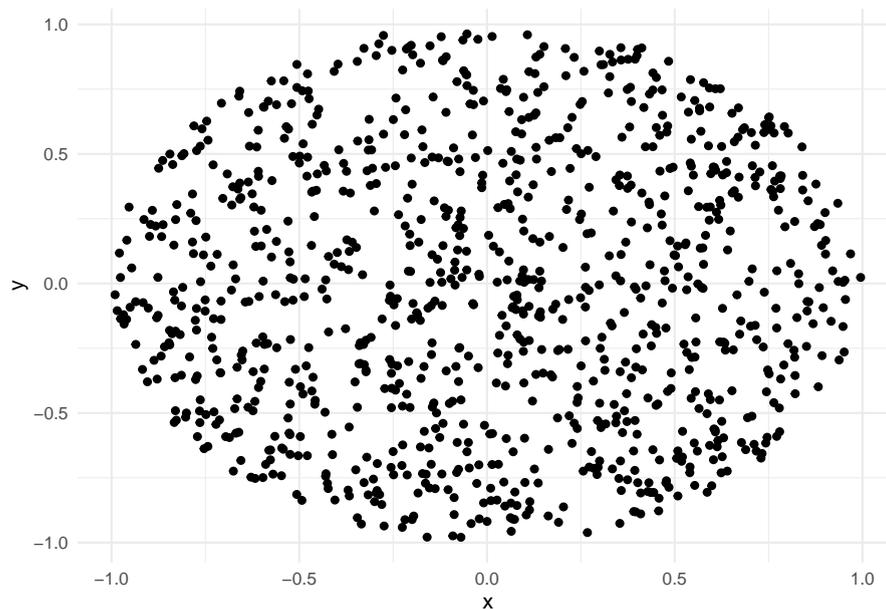


Figura 3.2: Simulación de 1000 puntos aleatorios distribuidos uniformemente sobre un disco de radio 1.

3.2. Técnicas básicas de simulación

Pasamos ya a la aplicación del conjunto de principios básicos más utilizados en simulación y que han sido descritos en detalle en el capítulo anterior, es decir, el método de la transformada inversa, el método de Aceptación-Rechazo y el método de composición. Para cada una de las técnicas se ha seleccionado un ejemplo práctico de utilidad.

3.2.1. Método de la transformada inversa

Aunque existen infinitas posibles aplicaciones de este método, en este caso se va a utilizar como ejemplo la distribución exponencial, un ejemplo clásico que fue abordado en la asignatura de simulación.

Variable exponencial

Para generar una variable exponencial mediante el método de la transformada inversa debemos tener en cuenta que su función de densidad es $f(x) = \lambda e^{-\lambda x}$ para $x > 0$ y, por tanto, su función de distribución es $F(x) = 1 - e^{-\lambda x}$. De esta forma el procedimiento consiste en:

1. Generar una variable aleatoria uniforme η en $[0, 1]$ mediante *runif*.
2. Como la inversa de F es $F^{-1}(u) = -\log(1 - u)/\lambda$. Tomamos $\xi = -\log(1 - \eta)/\lambda$. De esta forma, ξ se distribuye exponencialmente.

```
# Distribucion exponencial mediante método Transformada inversa
n <- 1000 # Número de puntos generados
lambda <- 0.5 # Parámetro de la exponencial

u <- runif(n) # Generamos variables aleatoria uniforme
x <- -log(1-u)/lambda # Generamos variable exp. mediante la función inversa

df <- data.frame(x = x) # Creamos un dataframe con los puntos

# Representamos gráficamente
ggplot(df, aes(x=x)) + geom_histogram(colour = "#1F3552", fill = "#4271AE") +
  theme_minimal()
```

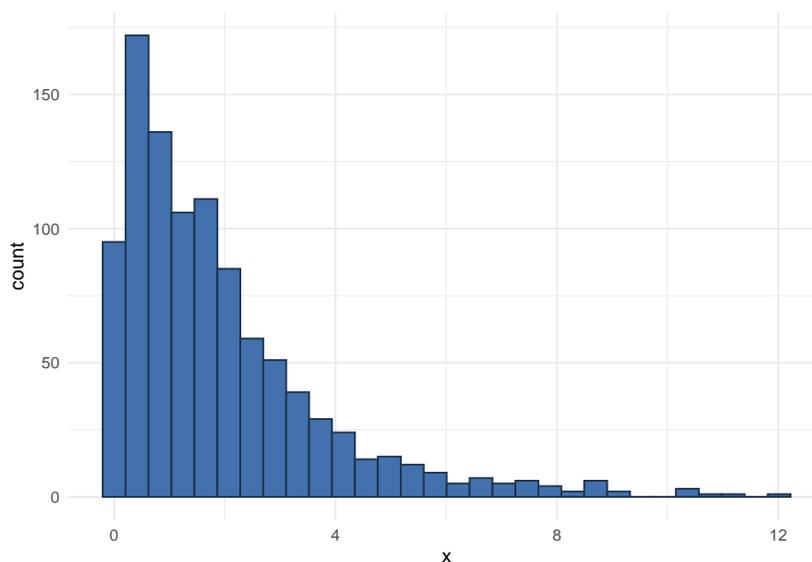


Figura 3.3: Simulación de la distribución exponencial mediante el método de la transformada inversa.

3.2.2. Método de Aceptación-Rechazo

Punto aleatorio uniformemente distribuido en una región cualquiera

Una aplicación sencilla del método de rechazo es utilizada para la generación de puntos aleatorios en una región cualquiera B contenida en otra región más simple C . En este caso, vamos a tomar B como un triángulo contenido en la caja unidad (región C). De esta forma, el algoritmo consiste en:

1. Generar un punto aleatorio en la caja unidad C tal y como hicimos en un ejemplo anterior.
2. Si está dentro de la región triangular B , se devuelve el punto. En caso contrario, se vuelve al paso 1.

```
# Punto aleatorio distribuido uniformemente en un triángulo
n <- 1000 # Número de puntos generados

# Definimos los límites de la región más simple (caja)
a <- c(0,1)
b <- c(0,1)
u <- runif(n, min = a[1],max = a[2]) # Generamos v.a. uni. como coordenada eje x
v <- runif(n, min = b[1],max = b[2]) # Generamos v.a. uni. como coordenada eje y

df <- data.frame(x =u,y=v) # Creamos dataframe con los puntos
# Seleccionamos aquellos puntos que caen en la región de interés
df <- df[which(df$x <= df$y),] # Es decir, cuando comp. x < comp. y

# Representamos gráficamente
ggplot(df, aes(x=x,y=y)) + geom_point()+theme_minimal()
```

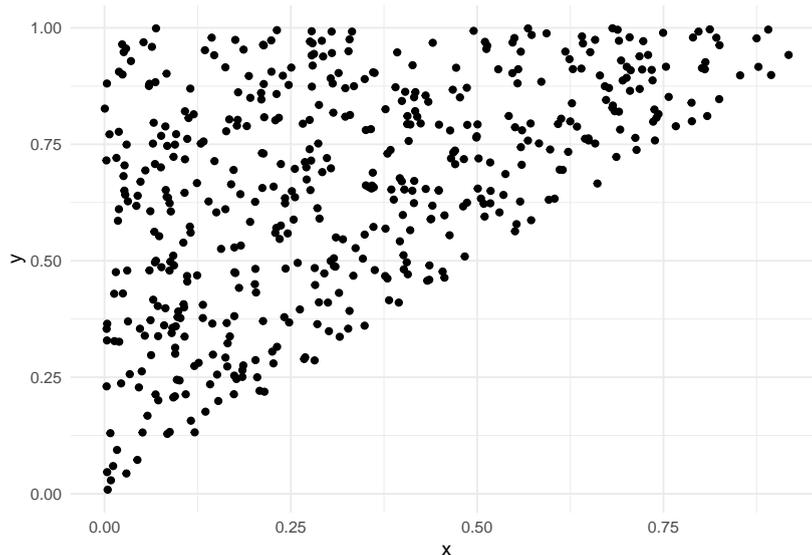


Figura 3.4: Simulación de puntos aleatorios uniformes generados en un triángulo mediante el método de rechazo.

Distribución normal

Queremos generar una variable aleatoria \mathbf{X} con distribución normal de parámetros $\mu = 0$ y $\sigma = 1$. Sabemos por tanto que su función de densidad es

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \forall x \in \mathbb{R}.$$

Para ello necesitamos otra función de densidad g que podamos simular de forma más sencilla, tal que exista una constante $M > 0$ que verifique $f(x) \leq Mg(x)$. En este caso elegimos la doble exponencial con parámetro $\lambda = 1$ y cota $M = \sqrt{\frac{2e}{\pi}}$. A continuación, se detalla el algoritmo entero.

1. Generamos la variable aleatoria doble exponencial \mathbf{Y} con función de densidad g . Para simplificar el proceso hacemos uso de la función `ddexp` del paquete `nimble`, que nos proporciona una secuencia de números distribuidos como una doble exponencial (en la siguiente sección se proporcionara un método sencillo para la generación de esta distribución).
2. A continuación, generamos independientemente una variable uniforme $\eta \sim Unif[0, M]$ donde M es la cota definida anteriormente.
3. Nos quedamos con aquellos valores que verifican $\eta \leq f(\mathbf{Y})/g(\mathbf{Y})$.

```
# Distribución normal mediante el método de Aceptación-Rechazo
library(nimble) # Cargamos la librería para generar la doble exponencial

n <- 1000 # Número de puntos generados

y <- rdexp(n) # Generamos la doble exponencial
M <- sqrt((2*exp(1))/pi) # Definimos la cota
u <- runif(n,min = 0,max = M) # Generamos la variable uniforme

# Una vez generado, nos quedamos solo con los valores que verifican
# la expresión dada
x <- y[u * ddexp(y) <= dnorm(y)]

df <- data.frame(x = x) # Creamos dataframe con los puntos

# Representamos gráficamente
ggplot(df, aes(x=x)) + geom_histogram(colour = "#1F3552", fill = "#4271AE") +
  theme_minimal()
```

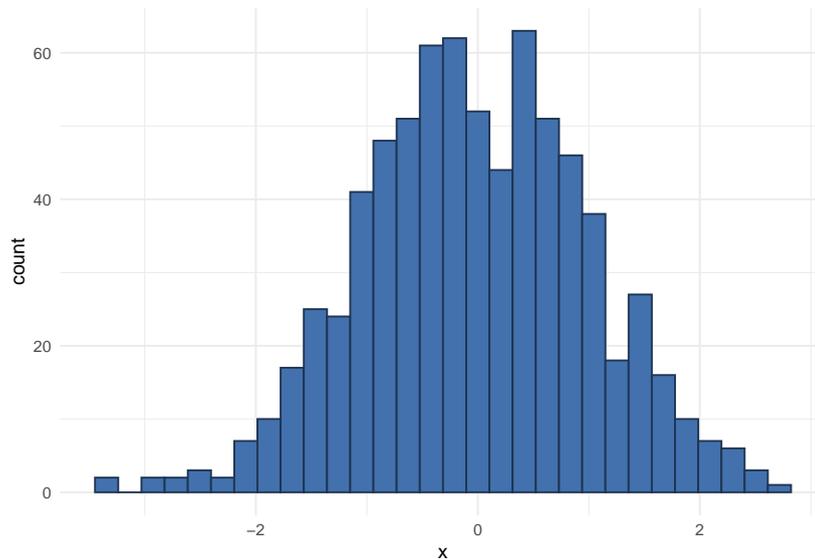


Figura 3.5: Simulación de una distribución normal mediante el método de aceptación-rechazo.

3.2.3. Método de composición

Doble exponencial

Un buen ejemplo para ilustrar el método de composición es la doble exponencial. Este algoritmo detallado a continuación es un caso particular de la mixtura de exponenciales vista en el curso de simulación. La función de densidad de la doble exponencial se escribe como

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}, \forall x \in \mathbb{R},$$

que se puede descomponer en

$$f(x) = \frac{1}{2}f_1(x) + \frac{1}{2}f_2(x)$$

donde f_1 y f_2 son exponenciales tal y como indicamos en la sección 2.2.3. Notad que en este caso p_1 y p_2 toman el valor $1/2$. De esta forma, el algoritmo para generar la variable X se puede escribir como:

1. Generamos dos variables uniformes $\eta_1, \eta_2 \sim Unif[0, 1]$.
2. Si $\eta_1 < 0,5$ (ya que $p_i = 1/2$), generamos la variable con densidad f_1 , es decir, $X = -\ln(1 - \eta_2)/\lambda$.
3. En caso contrario, generamos la variable con densidad f_2 , es decir, $X = \ln(\eta_2)$

```
# Doble exponencial mediante el método de composición
n <- 1000 # Número de variables generadas
lambda <- 0.5 # Parámetro de la doble exponencial

u <- runif(n) # Generamos variable aleatoria uniforme
v <- runif(n) # Generamos variable aleatoria uniforme
```

```
x <- ifelse(u < 0.5, -log(1-v)/lambda, log(v)/lambda) # Si u < 0.5 generamos f1
# En caso contrario f2

df <- data.frame(x = x) # Creamos dataframe con los puntos

# Representamos gráficamente
ggplot(df, aes(x=x)) + geom_histogram(colour = "#1F3552", fill = "#4271AE") +
  theme_minimal()
```

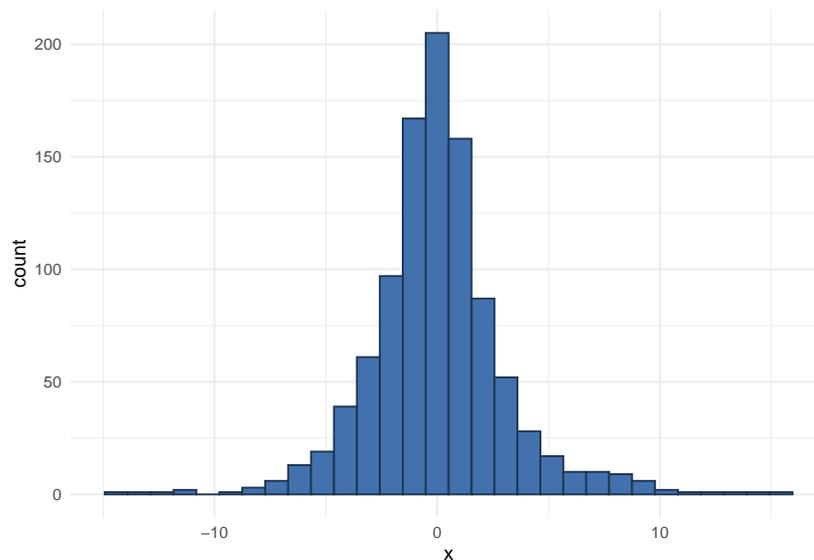


Figura 3.6: Simulación de la distribución doble exponencial mediante el método de composición.

3.2.4. Otros métodos

Box-Muller

A continuación, se desarrolla el algoritmo de Box-Muller para la generación de una distribución normal. Como se explicó en el Capítulo 2 ([15, Algoritmo 3.1 p. 54]), el proceso consiste en:

1. Generar $U_1 \sim \text{Unif}[0, 1]$ y tomamos $\theta = 2\pi U_1$.
2. Generar $U_2 \sim \text{Unif}[0, 1]$ y tomamos $\xi = -\ln U_2, R = \sqrt{2\xi}$.
3. Tenemos que $X = R \cos \theta$ y $Y = R \sin \theta$ son variables aleatorias normales independientes.

```
# Método Box-Muller para la generación de una distribución normal
n <- 1000 # Número de variables generadas

u <- runif(n) # Generamos variable aleatoria uniforme
v <- runif(n) # Generamos variable aleatoria uniforme
```

```

theta <- 2 * pi * u# Generamos theta
R <- sqrt(2 * (-log(v))) # Generamos R

df <- data.frame(x = R*cos(theta),y = R*sin(theta)) # Creamos dataframe

# Representamos gráficamente
ggplot(df, aes(x=x)) + geom_histogram(colour = "#1F3552", fill = "#4271AE") +
  theme_minimal()

```

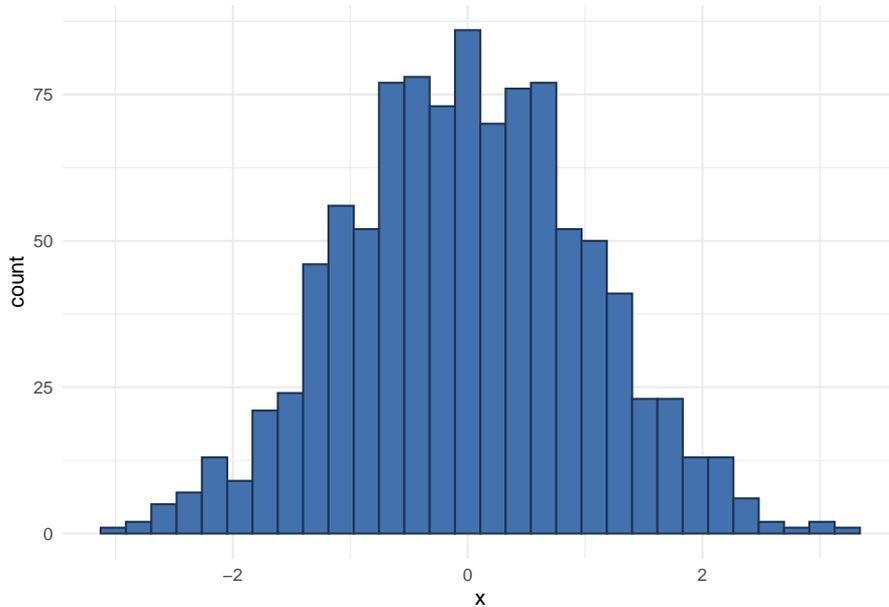


Figura 3.7: Simulación de la distribución normal mediante el método de Box-Muller.

3.3. Procesos de Poisson

Después de introducir los principios básicos de la simulación, pasamos ya a procesos puntuales más interesantes como son los procesos de Poisson. En esta sección, comenzaremos simulando procesos de Poisson homogéneos temporales y, a continuación, pasaremos a procesos espaciales, donde trataremos tanto procesos homogéneos como no homogéneos.

3.3.1. Proceso homogéneo temporal

En el caso unidimensional existen dos posibles enfoques a la hora de simular el proceso de Poisson tal y como vimos en el capítulo anterior. El primero de ellos se basa en generar los tiempos de espera entre llegadas, mientras que la idea del segundo es generar los tiempos de llegada. Ambas formas fueron abordadas en el curso de simulación.

Mediante tiempos de espera

En este caso, el hecho fundamental que hay que tener en cuenta es que los tiempos de espera entre sucesos se distribuyen como variables aleatorias exponenciales. Por tanto, para simular

([14, Algoritmo 3.5 p. 67]) una realización de un proceso de Poisson homogéneo en $[0, T_0]$:

1. Generamos n tiempos de espera S_1, S_2, \dots como variables aleatorias exponenciales de parámetro λ . Para ello utilizamos el método de la transformada inversa.
2. Calculamos los tiempos de llegada mediante la suma de los tiempos de espera, es decir, $T_n = \sum_{i=1}^n S_i$.
3. Devolvemos el conjunto de tiempos de llegada T_n que verifican $T_n < T_0$.

```
# Proceso de Poisson mediante tiempos de espera
T_0 <- 50 # Intervalo temporal
lambda <- 2 # Tasa de ocurrencia de los sucesos

n <- 200 # Número de tiempos de espera generados
u <- runif(n = n - 1) # Generamos n-1 variables aleatorias uniformes
t <- c(0, cumsum(- log(u) / lambda)) # Generamos tiempos de espera (exponencial)
# Y los sumamos para obtener los tiempos de llegada

df <- data.frame(t = t, y = 1:n) # Dataframe con los resultados
df <- df[df$t < T_0,] # Tomamos los tiempos de llegada < T_0

# Representación gráfica
ggplot(data=df, mapping=aes(x=t, y=y)) +
  geom_step()+
  theme_minimal()+ylab("Número de ocurrencias") + xlab("Tiempo")
```

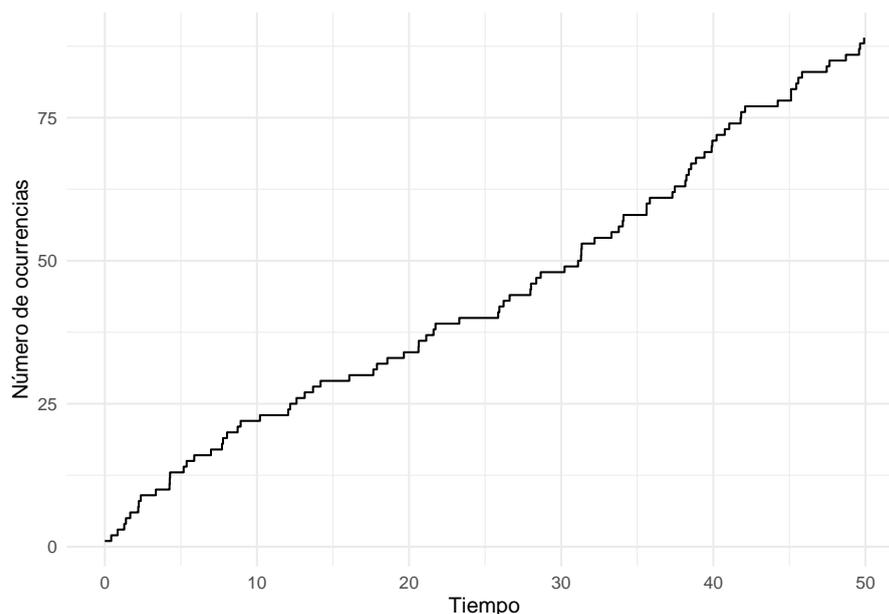


Figura 3.8: Simulación de un proceso de Poisson homogéneo unidimensional mediante la generación de los tiempos de espera.

Mediante generación condicional

En este caso, nos centramos en generar directamente los tiempos de llegada como variables aleatorias uniformes. Para simular ([14, Algoritmo 3.6 p. 67]) una realización de un proceso de Poisson homogéneo en $[0, T_0]$:

1. Generamos $N \sim Pois(\lambda T_0)$. Para ello utilizamos la función de R `rpois` que proporciona valores aleatorios de una distribución de Poisson con parámetro dado.
2. Dado $N = n$, generamos n variables i.i.d. $\eta_1, \eta_2, \dots, \eta_n \sim Unif[0, T_0]$.
3. Ordenamos $\{\eta_1, \dots, \eta_n\}$ para obtener los tiempos de llegada.

```
# Proceso de Poisson mediante generación condicional
T_0 <- 50 # Número de puntos
lambda <- 2 # Tasa de ocurrencia de los sucesos
# Generamos n° de ocurrencias en T_0 (como variable de Poisson)
npoints <- rpois(n = 1, lambda = lambda * T_0)

if(npoints > 0){
  t <- c(0, sort(runif(npoints - 1) * T_0)) # Ordenamos los tiempos de llegada
}else{
  t <- 0 # En el improbable caso de que npoints sea 0
}

df <- data.frame(y = 1:npoints, t = t) # Dataframe con los resultados

# Representación gráfica
ggplot(data=df, mapping=aes(x=t, y=y)) +
  geom_step()+
  theme_minimal()+ylab("Número de ocurrencias") + xlab("Tiempo")
```

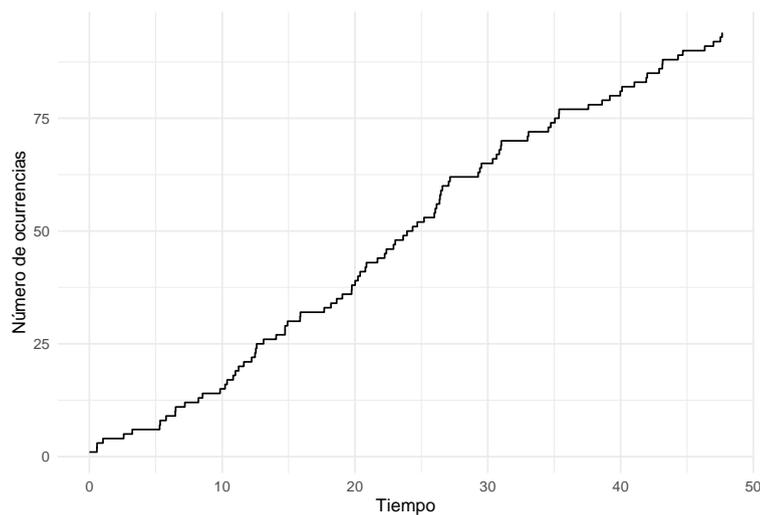


Figura 3.9: Simulación de un proceso de Poisson homogéneo mediante generación condicional.

3.3.2. Proceso espacial

Pasamos ahora a procesos de Poisson en dos dimensiones, dejando de lado la variable temporal y restringiéndonos a procesos puntuales espaciales únicamente.

Homogéneo

En este primer ejemplo tratamos el caso homogéneo y sobre una región rectangular (caja) $[0, T_x] \times [0, T_y]$ de área A visto en el curso de simulación. El procedimiento es prácticamente idéntico al ejemplo explicado para puntos aleatorios distribuido uniformemente en una caja. Y la generalización a regiones más complejas se lleva a cabo de forma idéntica a la planteada al inicio del capítulo. Así, en este caso:

1. Generamos $N \sim Pois(\lambda A)$ donde A es el área total. Para ello utilizamos nuevamente la función de R `rpois`.
2. Una vez dado el número de puntos $N = n$, generamos n variables i.i.d. $x_1, x_2, \dots, x_n \sim Unif[0, T_x]$ y otras n $y_1, y_2, \dots, y_n \sim Unif[0, T_y]$
3. Tomamos los puntos aleatorios como (x_i, y_i) .

```
# Proceso de Poisson homogéneo espacial
# Definimos la región (caja)
T_x <- 1
T_y <- 1
A <- T_x * T_y # Area total

lambda <- 100 # Tasa de ocurrencia de los sucesos
npoints <- rpois(n = 1, lambda = lambda * A) # Generamos el número de ocurrencias

# Generamos los puntos aleatorios
x <- runif(npoints) * T_x
y <- runif(npoints) * T_y

df <- data.frame(x = x, y = y) # Dataframe con los resultados

# Representación gráfica
ggplot(data=df, mapping=aes(x=x, y=y)) +
  geom_point()+ theme_minimal()
```

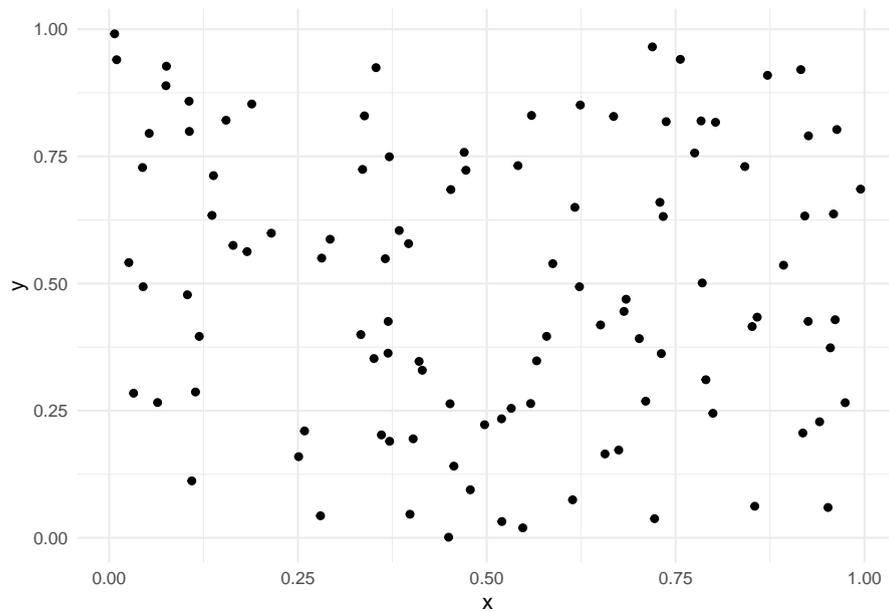


Figura 3.10: Proceso de Poisson homogéneo espacial mediante generación condicional.

No homogéneo: λ depende de la coordenada x

En el caso de que el proceso de Poisson no sea homogéneo, debemos de tener en cuenta la función de distribución de los puntos sobre el espacio. En este ejemplo vamos a simular un proceso de Poisson cuya intensidad λ depende del eje x , es decir, $\lambda(x, y) = x^2$. Para ello:

1. Comenzamos generando el número de puntos aleatorios como una variable de Poisson, es decir, $N \sim Pois(\lambda T_0)$. El intervalo $[0, T_0]$ define el segmento de la componente y .
2. Una vez dado el número de puntos $N = n$, generamos n variables i.i.d. $y_1, y_2, \dots, y_n \sim Unif[0, T_0]$ ya que la componente y no depende de λ .
3. A continuación, generamos las componente x teniendo en cuenta cómo se distribuyen. En este caso, tomamos $f(x) = x^2$, por lo que mediante el método de la transformada inversa generamos x_1, x_2, \dots, x_n .
4. Tomamos los puntos aleatorios como (x_i, y_i) .

```
# Proceso de Poisson no homogéneo (lambda depende de x)
# Definimos la región
T_x <- 1
T_y <- 1
A <- T_x * T_y # Area total

lambda <- 1000 # Tasa de ocurrencia de los sucesos
npoints <- rpois(n = 1, lambda = lambda * A) # Generamos el número de ocurrencias

# Generamos los puntos aleatorios
x <- sqrt(runif(npoints)) * T_x # Modificamos distribución componente x
```

```

y <- runif(npoints) * T_y

df <- data.frame(x = x, y = y) # Dataframe con los resultados

# Representación gráfica
ggplot(data=df, mapping=aes(x=x, y=y)) +
  geom_point()+
  theme_minimal()

```

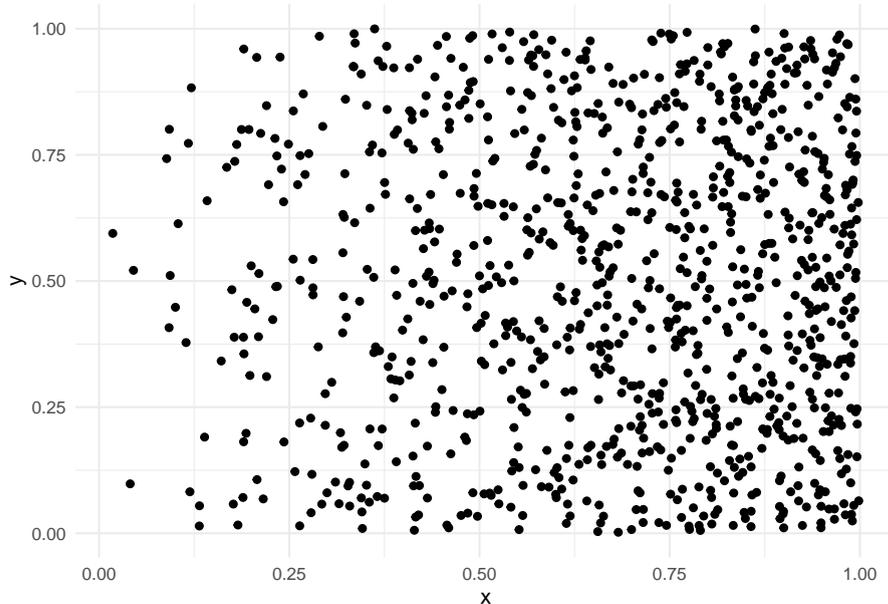


Figura 3.11: Simulación de una realización de un proceso de Poisson no homogéneo espacial donde λ depende de x .

No homogéneo en $B_r^2(o)$ mediante el procedimiento radial

Tal y como se vió en el capítulo anterior, existe un método para simular un proceso de Poisson en un bola $B_r^2(o)$ de radio r utilizando los tiempos de espera. El algoritmo ([44]) se expone a continuación.

1. Generamos los tiempos de espera entre sucesos S_1, S_2, \dots como variables exponenciales de parámetro $\lambda V(B_r^d(o))$. Donde $V(B_r^d(o))$ es en este caso ($d = 2$) el área del disco de radio r que viene dada por πr^2 .
2. Generamos variables i.i.d. $\mathbf{U}_1, \mathbf{U}_2, \dots \sim Unif(B_r^2(o))$. Para este utilizamos la técnica descrita en la Sección 3.1.
3. A continuación, calculamos las distancias de los puntos al origen a partir de los tiempos de espera. Es decir, $R_0 = 0$ y $R_i = R_{i-1} + S_i$, con $i = 1, 2, \dots$.
4. Los puntos simulados se obtienen multiplicando $R_1 \mathbf{U}_1, R_2 \mathbf{U}_2, \dots$. Esta transformación escala la abscisa y la ordenada de cada punto uniforme generado, rompiendo así la homogeneidad del proceso inicial de Poisson.

```

# Procedimiento radial
lambda <- 1000 # Tasa de ocurrencia de los sucesos
r <- 1 #Definimos el radio del disco
npoints <- rpois(n = 1, lambda = lambda * pi * (r^2)) # Generamos n° ocurrencias

# Generamos los puntos en el bola
u <- runif(npoints) #Variable aleatoria uniforme
v <- runif(npoints) #Variable aleatoria uniforme
x <- r * sqrt(u) * cos(2 * pi * v)
y <- r * sqrt(u) * sin(2 * pi * v)

s <- rexp(npoints,lambda * pi*(r^2)) # Generamos los tiempos de espera
R <- cumsum(s) # Obtenemos los radios (distancias al origen)

df <- data.frame(x = R*x,y = R*y)# Dataframe con los resultados

# Representación gráfica
ggplot(df, aes(x=x,y=y)) +
  geom_point() + theme_minimal()

```

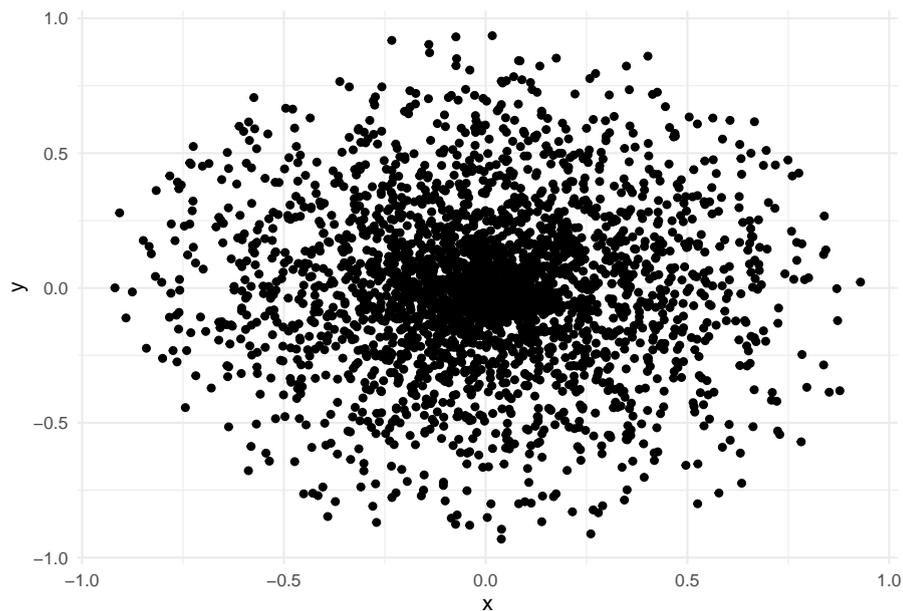


Figura 3.12: Simulación de una realización de un proceso de Poisson no homogéneo espacial mediante el procedimiento radial.

En la Figura 3.12 se aprecia claramente que se trata de un proceso de Poisson no homogéneo ya que la densidad de puntos es mucho mayor alrededor del origen. Esto se puede visualizar de forma muy clara representando la función de densidad de una de las componentes.

```
ggplot(df, aes(x=x)) +
  geom_density() + theme_minimal()
```

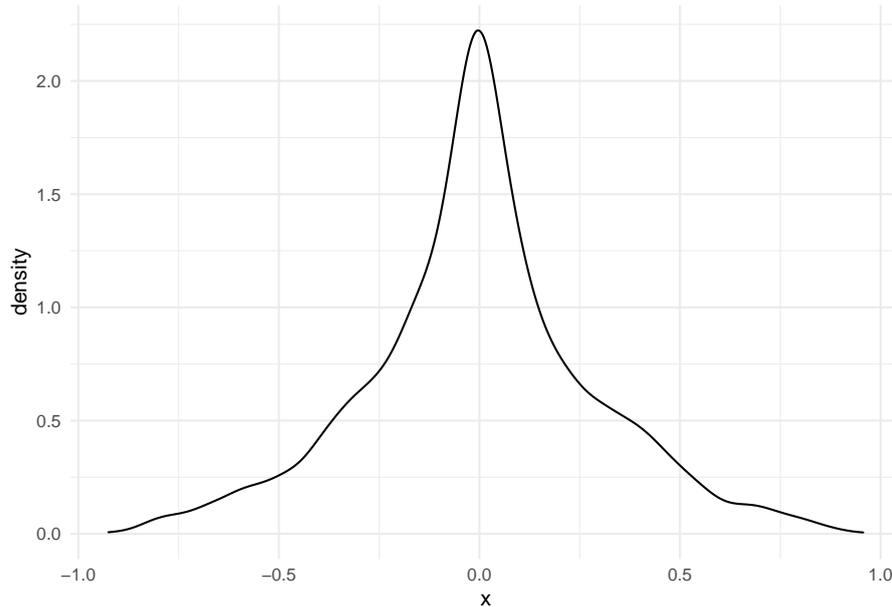


Figura 3.13: Función de densidad de la componente x del proceso de Poisson generado mediante el procedimiento radial.

En la Figura 3.13 se observa como la componente x del proceso de Poisson generado se distribuye como una doble exponencial, lo cual confirma la no homogeneidad del proceso.

No homogéneo: λ proporcional a la función de densidad normal

Otro ejemplo de proceso de Poisson no homogéneo interesante es aquel en el que la función λ es proporcional a la función de densidad de una distribución normal en el plano. Por simplificar, tomaremos el caso en el que la media es cero, la matriz de covarianzas es la identidad y no existe correlación entre componentes. De esta forma, la función $\lambda(x, y)$ se puede escribir como:

$$\lambda(x, y) = f_{XY}(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{-\frac{x^2-2xy+y^2}{2(1-\rho^2)}}$$

donde $\rho \in (-1, 1)$ es la correlación entre X e Y que en este caso simplificaremos a $\rho = 0$.

El proceso para simular una realización se explica a continuación:

1. Comenzamos generando el número de puntos aleatorios como una variable de Poisson, es decir, $N \sim Pois(\lambda_0)$.
2. Una vez dado el número de puntos $N = n$, generamos n variables i.i.d. y_1, y_2, \dots, y_n con distribución normal.
3. A continuación, generamos las componente x . Generamos n variables i.i.d. x_1, x_2, \dots, x_n con distribución normal.

4. Tomamos los puntos aleatorios como (x_i, y_i) .

```
# Proceso de Poisson no homogéneo (lambda depende de x)
lambda <- 1000 # Tasa de ocurrencia de los sucesos
npoints <- rpois(n = 1, lambda = lambda) # Generamos el número de ocurrencias

# Generamos los puntos como variables normales independientes
x <- rnorm(npoints)
y <- rnorm(npoints)

df <- data.frame(x = x, y = y) # Dataframe con los resultados

# Representación gráfica
ggplot(data=df, mapping=aes(x=x, y=y)) +
  geom_point()+
  theme_minimal()
```

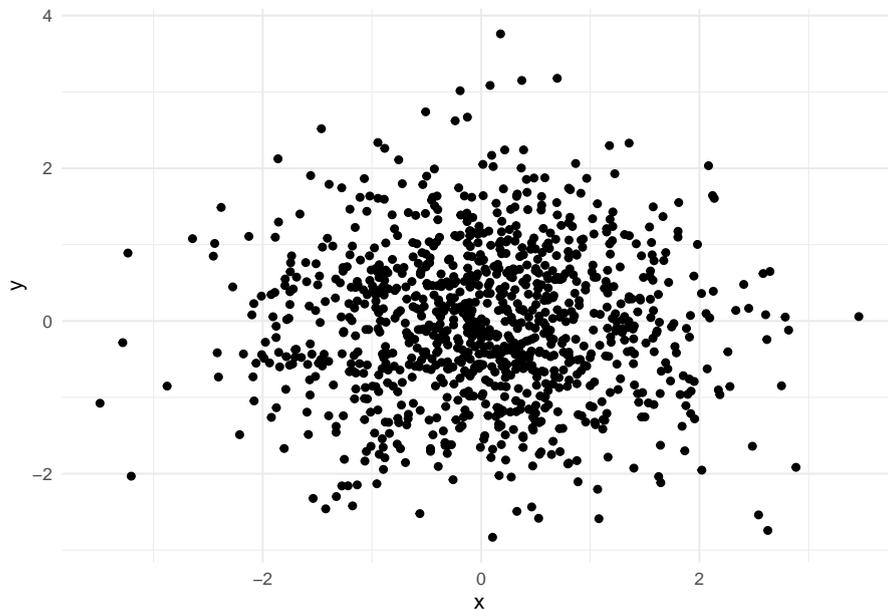


Figura 3.14: Simulación de una realización de un proceso de Poisson no homogéneo espacial donde λ coincide con la densidad conjunta de dos variables aleatorias normales independientes.

En la Figura 3.14 se aprecia que los puntos no están distribuidos de forma homogénea, habiendo mayor densidad alrededor del origen. Igual que en el ejemplo anterior, podemos representar la distribución de los puntos en una de las componentes (eje x por ejemplo) para apreciar como los puntos se distribuyen según la función de densidad de una normal, tal y como queríamos (Figura 3.15).

```
ggplot(df, aes(x=x)) +
  geom_density() + theme_minimal()
```

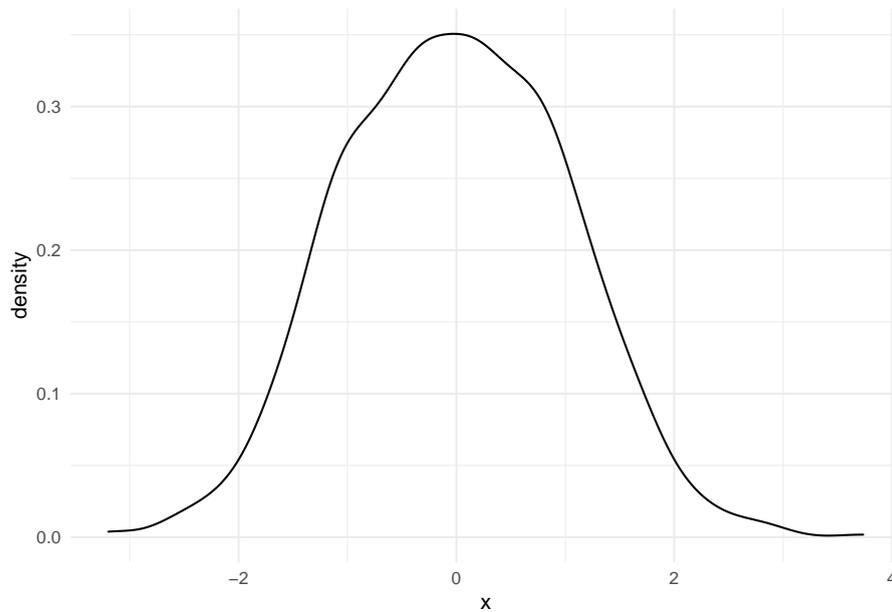


Figura 3.15: Función de densidad de la componente x de un proceso de Poisson con λ proporcional a la función de densidad de una normal.

3.4. Procesos derivados del proceso de Poisson

Finalmente, en esta sección presentamos los algoritmos para la generación de procesos derivados del proceso de Poisson, concretamente, los procesos de Cox y procesos Clúster de Poisson.

3.4.1. Proceso de Cox

A continuación, presentamos dos aplicaciones del algoritmo [14, Algoritmo 3.10 p. 91] para generar procesos de Cox.

Λ variable aleatoria exponencial

En este primer ejemplo, tomamos la intensidad Λ como una variable aleatoria con distribución exponencial. Simulamos varias realizaciones en el plano para que se aprecien mejor las diferencias entre ellas debido al carácter aleatorio de la intensidad. El procedimiento es similar al utilizado en el proceso de Poisson homogéneo en una caja solo que en este caso la intensidad toma un valor aleatorio en cada realización:

1. Generamos la variable intensidad $\Lambda \sim Exp(\alpha)$. Para ello utilizamos el comando de R *rexp*.
2. Generamos $N \sim Pois(\Lambda A)$ donde A es el área total de la región en el plano. Para ello utilizamos nuevamente la función de R *rpois*.
3. Una vez dado el número de puntos $N = n$, generamos n variables i.i.d. $x_1, x_2, \dots, x_n \sim Unif[0, T_0]$ y otras n $y_1, y_2, \dots, y_n \sim Unif[0, T_0]$ donde T_0 es el lado de la caja con área A .
4. Tomamos los puntos aleatorios como (x_i, y_i) . Este proceso se puede repetir tantas veces como queramos para obtener diferentes realizaciones.

```

# Proceso de Cox: Lambda variable exponencial
# Definimos la región donde se localizan los puntos
T_x <- 100
T_y <- 100
A <- T_x * T_y # Area total

# Simulamos tres realizaciones
for (i in 1:3) {

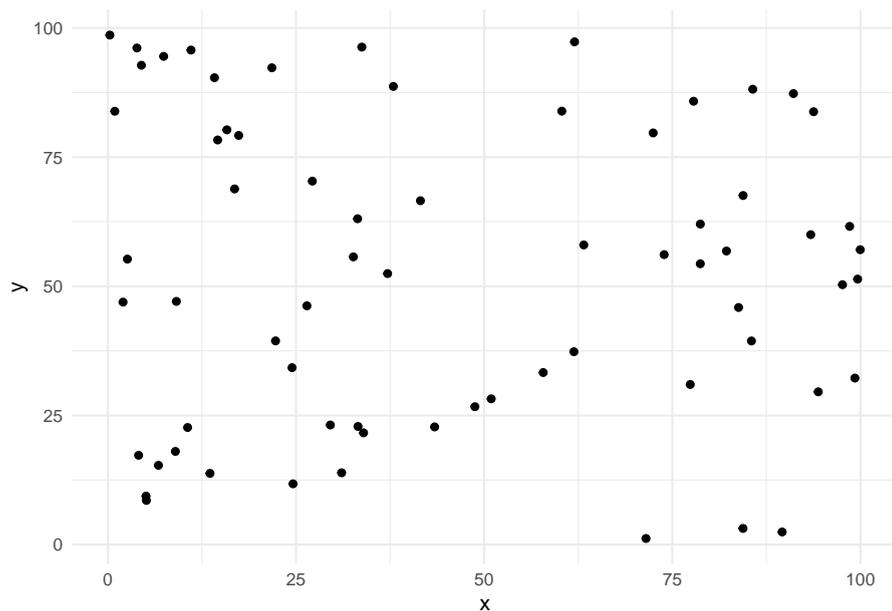
  # Generamos lambda como una variable exponencial de parámetro 30
  lambda <- rexp(1, rate = 30)

  npoints <- rpois(n = 1, lambda = lambda * A) # Generamos n^o ocurrencias
  x <- runif(npoints) * T_x
  y <- runif(npoints) * T_y

  df <- data.frame(x = x, y = y) # Dataframe con los resultados

  # Representación gráfica
  p <- ggplot(data=df, mapping=aes(x=x, y=y)) +
    geom_point() + theme_minimal()
  print(p)
}

```



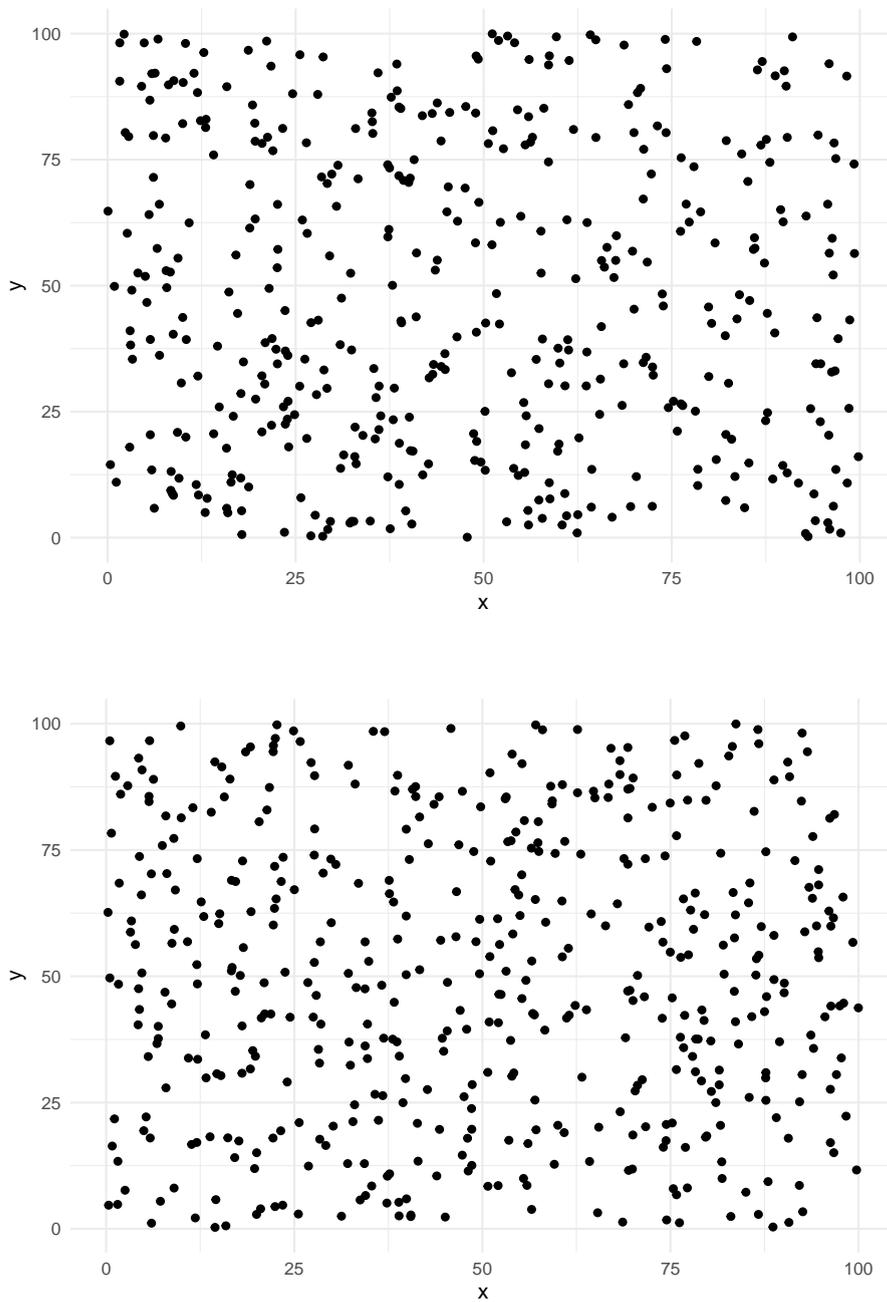


Figura 3.16: Tres realizaciones de un proceso de Cox con Λ variable exponencial de parámetro 30.

Proceso Log-Gaussiano

Otro proceso de Cox usualmente estudiado es el Log-Gaussiano. Ya que una variable aleatoria normal puede tomar valores negativos, no es posible utilizarla directamente para generar la intensidad Λ por ésta debe ser siempre positiva. Para superar esta dificultad se utiliza la distribución lognormal o Log-Gaussiana. Una variable aleatoria X se dice que sigue una distribución lognormal $X \sim \text{Lognormal}(\mu, \sigma^2)$ si el logaritmo de X es una variable normal con media μ y desviación típica σ .

El algoritmo para generar este tipo de procesos es idéntico al ejemplo anterior, solo que debemos de usar la distribución lognormal para generar Λ . De esta forma, el procedimiento queda como:

1. Generamos la variable intensidad $\Lambda \sim \text{Lognormal}(\mu, \sigma^2)$. Para ello utilizamos el comando de R `rnorm`.
2. Procedemos igual que en el ejemplo anterior. Generamos $N \sim \text{Pois}(\Lambda A)$ donde A es el área total de la región en el plano.
3. Una vez dado el número de puntos $N = n$, generamos n variables i.i.d. $x_1, x_2, \dots, x_n \sim \text{Unif}[0, T_0]$ y otras n $y_1, y_2, \dots, y_n \sim \text{Unif}[0, T_0]$ donde T_0 es el lado de la caja con área A .
4. Tomamos los puntos aleatorios como (x_i, y_i) . Este proceso se puede repetir tantas veces como queramos para obtener diferentes realizaciones.

```
# Proceso Log-Gaussiano de Cox
# Definimos la región donde se localizan los puntos
T_x <- 10
T_y <- 10
A <- T_x * T_y # Area total

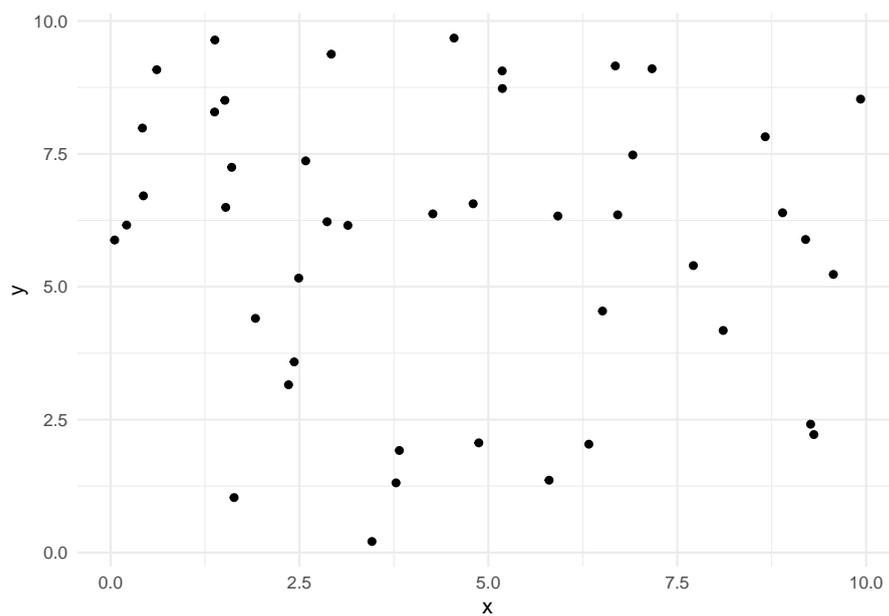
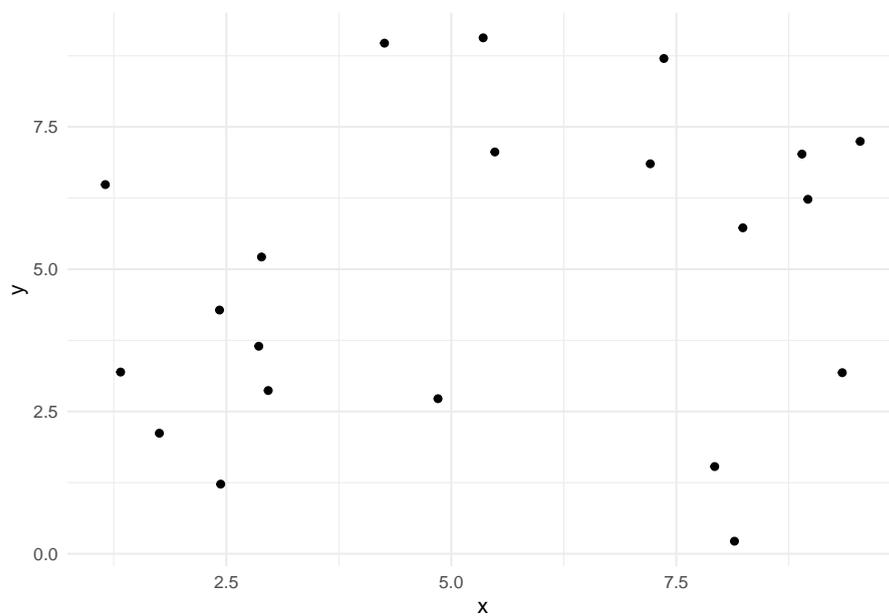
# Simulamos tres realizaciones
for (i in 1:3) {

  # Generamos lambda como la exponencial de una variable aleatoria normal
  lambda <- exp(rnorm(1))

  npoints <- rpois(n = 1, lambda = lambda * A) # Generamos nº ocurrencias
  x <- runif(npoints) * T_x
  y <- runif(npoints) * T_y

  df <- data.frame(x = x, y = y) # Dataframe con los resultados

  # Representación gráfica
  p <- ggplot(data=df, mapping=aes(x=x, y=y)) +
    geom_point()+theme_minimal()
  print(p)
}
```



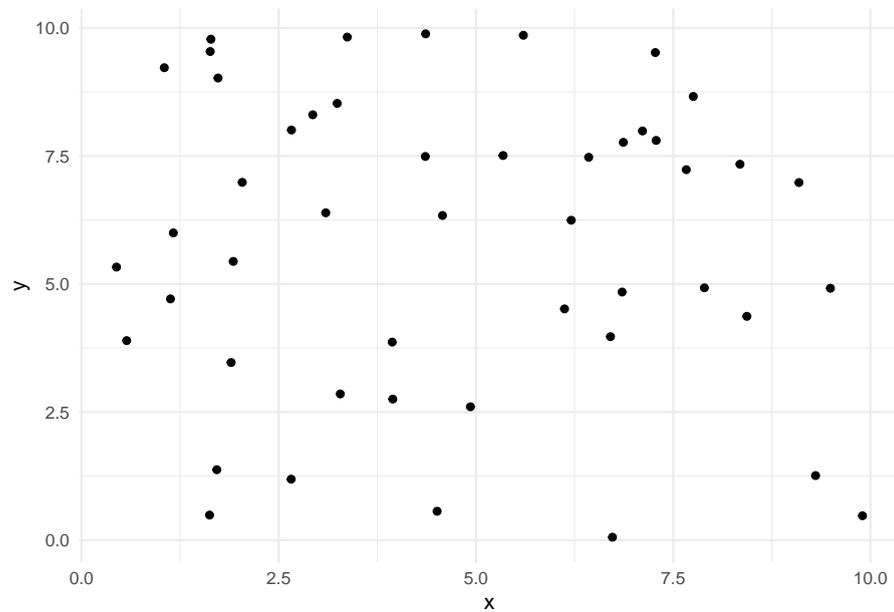


Figura 3.17: Tres realizaciones de un proceso Log-Gaussiano de Cox.

3.4.2. Proceso Clúster de Poisson

Clústers acotados

Para generar un proceso Clúster de Poisson debemos comenzar generando los clústers y , a continuación, simular los puntos aleatorios asociados a cada clúster. En este caso vamos a simular un proceso clúster espacial. Este procedimiento sigue la idea del desarrollado en [14, Algoritmo 3.11 p. 91], consiste en:

1. Comenzamos generando una realización de un proceso de Poisson espacial homogéneo tal y como describimos en la sección anterior (conviene decir que el proceso sería semejante si se tratara de un proceso no homogéneo). Este primer proceso puntual nos proporciona los padres o clústers a partir de los cuales generamos los puntos aleatorios. Es decir, generamos una realización de x del proceso de Poisson homogéneo con intensidad κ en la región W .
2. Una vez generados los clústers, procedemos a generar procesos de Poisson para cada uno de los clústers. Para ello, para cada $x_i \in x$ generamos una realización de ζ_{x_i} , es decir, generamos un proceso de Poisson centrado en el disco $B_r(x_i)$ de radio r .
3. Una vez repetido el paso anterior para todos los clústers generados en el paso 1 se acaba el proceso tomando todos los puntos generados que caen dentro de la región W , es decir, $y = W \cap \bigcup_i \zeta_{x_i}$.

```
# Proceso Clúster
# Comenzamos generando un proceso de Poisson homogéneo espacial en una caja
T_x <- 1
T_y <- 1
A <- T_x * T_y # Area total
```

```

lambda <- 5 # Parámetro de ocurrencia de los sucesos
npoints <- rpois(n = 1, lambda = lambda * A) # Generamos n° de ocurrencias en T_0

x <- runif(npoints) * T_x
y <- runif(npoints) * T_y
df <- data.frame(x = x, y = y) # Dataframe con los resultados
# Representación gráfica
ggplot(data=df, mapping=aes(x=x, y=y)) +
  geom_point()+
  theme_minimal()

```

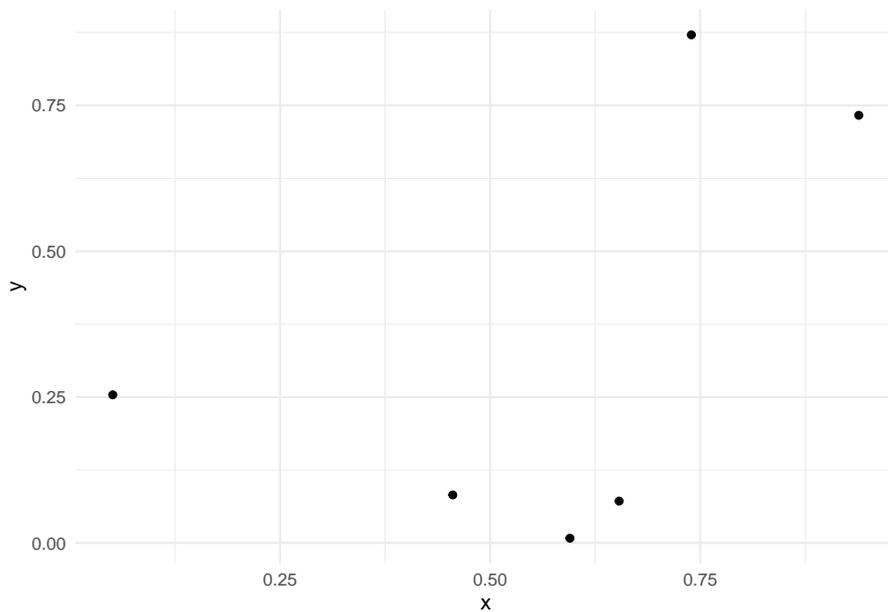


Figura 3.18: Simulación de los puntos padre o clústers a partir de los cuales se simularán los puntos aleatorios.

```

# Para cada punto generado, generamos un nuevo clúster de puntos
df2 <- as.data.frame(c())
for (i in 1:dim(df)[1]) {

  n <- rpois(n = 1, lambda = 5) # Número de puntos generados en cada clúster
  r <- 0.05 #Definimos el radio del disco

  u <- runif(n) #Variable aleatoria uniforme
  v <- runif(n) #Variable aleatoria uniforme

  #Aplicamos la transformación a polares
  x <- r * sqrt(u) * cos(2*pi * v) + df$x[i]
  y <- r * sqrt(u) * sin(2*pi * v) + df$y[i]
}

```

```

df2 <- rbind(df2,data.frame(x = x,y = y))
}

df <- rbind(df,df2) # Unimos los puntos padre con los clústers

ggplot(data=df, mapping=aes(x=x, y=y)) +
  geom_point()+
  theme_minimal()

```

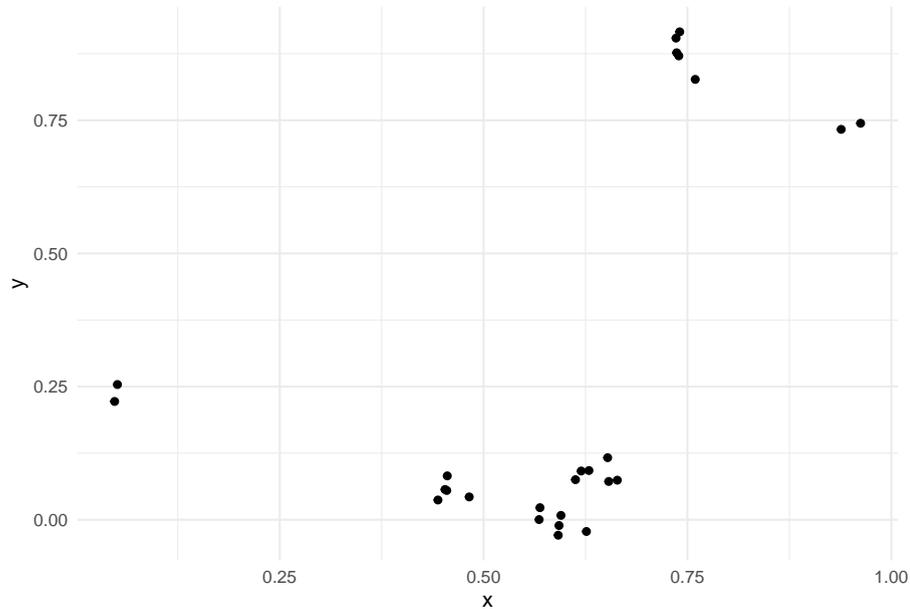


Figura 3.19: Simulación de una realización de un proceso Clúster de Poisson.

Clústers no acotados

El ejemplo anterior se puede complicar añadiendo una fuente de aleatoriedad más en el radio del disco, es decir, podemos tomar los radios de los discos como una variable aleatoria de forma que cada disco tenga radio R_i . Así, obtenemos el proceso de clústers no acotado. Este algoritmo se desarrolla en detalle en [14, Algoritmo 3.12 p. 92], aquí presentamos una versión simplificada.

1. Generamos el número de puntos $N \sim Pois(\lambda A)$, donde A es la región donde pueden situarse los puntos. En nuestro caso, una caja.
2. Generamos los puntos padre x_1, x_2, \dots, x_n en dicha caja.
3. Generamos $N = n$ radios R_i de los discos de los clústers como una variable aleatoria exponencial.
4. Para cada clúster i generamos el número de puntos $N_i \sim Pois(\mu \nu_d(B_{R_i}(x_i)))$ (donde $\nu_d(B_{R_i}(x_i))$ es el área del disco) y los n_i puntos en su disco correspondiente $B_{R_i}(x_i)$
5. Se combinan todos los puntos generados y se descartan aquellos que caen fuera de la región A .

```

# Proceso Clúster de Poisson no acotado
# Comenzamos generando un proceso de Poisson ordinario
T_x <- 1
T_y <- 1
A <- T_x * T_y # Area total

lambda <- 5 # Parámetro de ocurrencia de los sucesos
npoints <- rpois(n = 1, lambda = lambda * A) # Generamos n° ocurrencias

x <- runif(npoints) * T_x
y <- runif(npoints) * T_y
df <- data.frame(x = x, y = y) # Dataframe con los resultados

# Representación gráfica
ggplot(data=df, mapping=aes(x=x, y=y)) +
  geom_point()+
  theme_minimal() + xlim(0,1) + ylim(0,1)

```

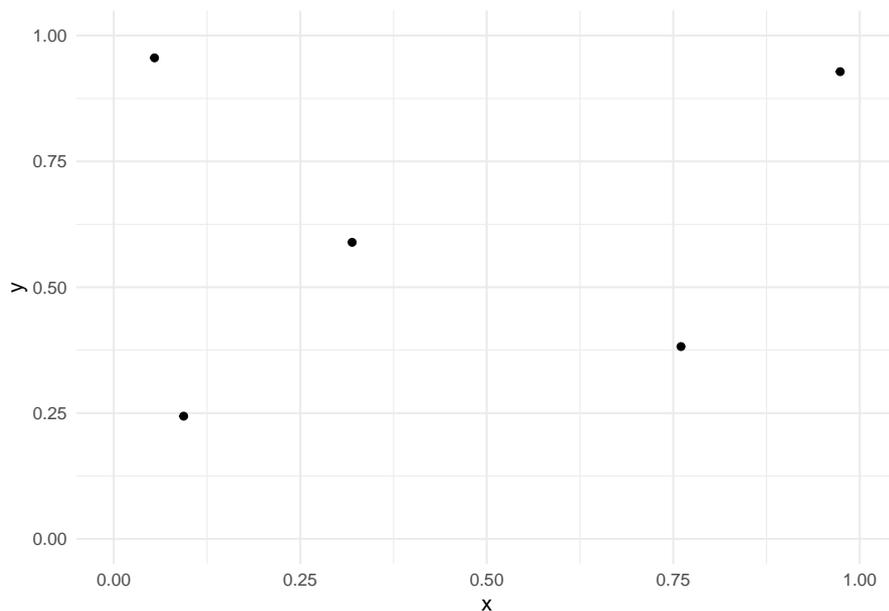


Figura 3.20: Simulación de los puntos padre o clústers a partir de los cuales se simularán los puntos aleatorios.

```

# Para cada punto generado, generamos un nuevo clúster de puntos
df2 <- as.data.frame(c())
mu <- 10 # Parámetro intensidad de los clústers
for (i in 1:dim(df)[1]) {

  r <- rexp(1,10) # Definimos el radio del disco como una variable exponencial
  n <- rpois(n = 1, lambda = mu * pi * r) # N° puntos generados en cada clúster

```

```

u <- runif(n) #Variable aleatoria uniforme
v <- runif(n) #Variable aleatoria uniforme

#Aplicamos la transformación a polares
x <- r * sqrt(u) * cos(2*pi * v) + df$x[i]
y <- r * sqrt(u) * sin(2*pi * v) + df$y[i]

df2 <- rbind(df2,data.frame(x = x,y = y))
}

df <- rbind(df,df2) # Unimos los puntos padre con los clústers
df <- df[df$x <=T_x & df$x >=0 &
        df$y <=T_y & df$y >=0,]

ggplot(data=df, mapping=aes(x=x, y=y)) +
  geom_point()+
  theme_minimal() + xlim(0,1) + ylim(0,1)

```

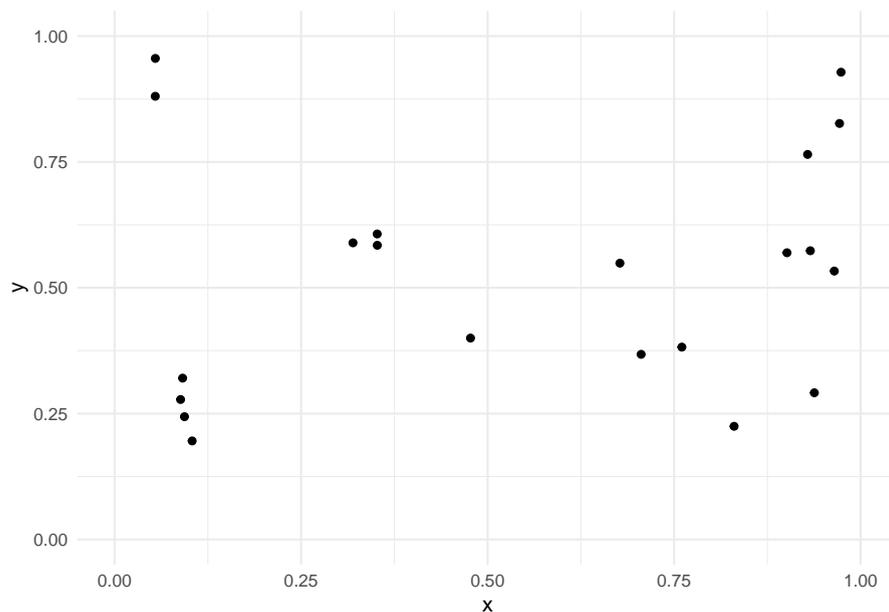


Figura 3.21: Simulación de una realización de un proceso Clúster de Poisson donde los radios de los discos son variables aleatorias.

Conclusiones

Este Trabajo Fin de Máster (TFM), titulado, *Una Breve Introducción a la Simulación de Patrones Puntuales Espaciales* aborda, utilizando técnicas estándar en simulación, basadas en condicionamiento y cambio de variable, la generación de patrones puntuales, esencialmente, bajo el marco de procesos de Poisson, centrándose, fundamentalmente, en la formulación espacial de dichos procesos. Más concretamente, tras introducir, en el Capítulo I, los elementos básicos y definiciones que permiten describir la clase de procesos puntuales, bajo el marco de procesos de Poisson, en el Capítulo 2, se realiza una breve revisión de las herramientas básicas que intervienen en los métodos de simulación estocástica de distribuciones unidimensionales y bidimensionales.

Específicamente, se adoptarán algoritmos de simulación, basados en dichos métodos, que permiten, a partir de la aplicación de las herramientas usuales de condicionamiento, o bien, del Teorema de Cambio de Variable, la generación sencilla de modelos de Poisson homogéneos, no homogéneos y doblemente estocásticos en el espacio. Se parte, en primera instancia, de la generación básica de estos procesos en el tiempo (en parte analizada en el curso de simulación), donde la generación, por ejemplo, de procesos de Poisson homogéneos en el tiempo, se obtiene de forma directa, a partir de la propiedad de incrementos independientes y estacionarios de dicho proceso, mediante aplicación de las técnicas tradicionales de generación de variables aleatorias unidimensionales. Las diferentes caracterizaciones de un proceso puntual, o bien, en nuestro caso, de un proceso de Poisson en sus diferentes variantes, en términos de una función aleatoria de recuento (o distribución aleatoria), o bien, en términos de los tiempos de espera, así como de los tiempos de ocurrencia asociados a los eventos aleatorios analizados, juegan un papel fundamental en el diseño de dichos métodos según se describe en los Capítulos 1–2 del presente TFM. Finalmente, el Capítulo 3 ilustra la implementación de los métodos de simulación estocástica descritos en el Capítulo 2, en código R, que permiten, en particular, graficar las salidas deseadas, incluyendo realizaciones de patrones puntuales y funciones de densidad.

Debemos resaltar, según se comenta, en la Introducción y en los Capítulos 1–2 de este TFM, la importancia y utilidad de los métodos de simulación descritos, en la validación de modelos y técnicas de predicción para el análisis de eventos que ocurren de forma aleatoria en el tiempo o espacio. De hecho, como líneas abiertas que han quedado pendientes para trabajar en este TFM, destacaremos:

- Revisión de los métodos de generación aleatoria de patrones puntuales espacio–temporales, como extensión inmediata de los métodos analizados en el caso espacial.
- Aplicación de las técnicas de simulación estocástica descritas al análisis estadístico de patrones temporales, espaciales y espacio–temporales.
- Aplicación de los métodos de generación descritos en el análisis asintótico de los métodos de estimación paramétrica por máxima verosimilitud para patrones puntuales.
- Aplicación de los métodos de generación descritos en el análisis asintótico de los métodos de estimación para procesos puntuales basados en estadísticos de momentos.

- Aplicación a datos reales de las técnicas de estimación usuales, testadas en el punto anterior, y validación posterior de los métodos de estimación y predicción *plug-in* asociados.

Bibliografía

- [1] D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Probability and Its Applications. Springer New York, 2013.
- [2] Peter J Diggle. *Statistical analysis of spatial and spatio-temporal point patterns*. CRC press, 2013.
- [3] Jesper Moller and Rasmus Plenge Waagepetersen. *Statistical inference and simulation for spatial point processes*. CRC Press, 2003.
- [4] Janine Illian, Antti Penttinen, Helga Stoyan, and Dietrich Stoyan. *Statistical analysis and modelling of spatial point patterns*, volume 70. John Wiley & Sons, 2008.
- [5] David R Cox. Some statistical methods connected with series of events. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17(2):129–157, 1955.
- [6] Jan Grandell. *Doubly stochastic Poisson processes*, volume 529. Springer, 2006.
- [7] Sung Nok Chiu, Dietrich Stoyan, Wilfrid S Kendall, and Joseph Mecke. *Stochastic geometry and its applications*. John Wiley & Sons, 2013.
- [8] Jonatan A González, Francisco J Rodríguez-Cortés, Ottmar Cronie, and Jorge Mateu. Spatio-temporal point process statistics: a review. *Spatial Statistics*, 18:505–544, 2016.
- [9] Yongtao Guan, Rasmus Waagepetersen, and Colin M Beale. Second-order analysis of inhomogeneous spatial point processes with proportional intensity functions. *Journal of the American Statistical Association*, 103(482):769–777, 2008.
- [10] Abdollah Jalilian, Yongtao Guan, and Rasmus Waagepetersen. Decomposition of variance for spatial cox processes. *Scandinavian Journal of Statistics*, 40(1):119–137, 2013.
- [11] Rasmus Waagepetersen and Yongtao Guan. Two-step estimation for inhomogeneous spatial point processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):685–702, 2009.
- [12] Rasmus Waagepetersen, Yongtao Guan, Abdollah Jalilian, and Jorge Mateu. Analysis of multispecies point patterns by using multivariate log-gaussian cox processes. *Journal of the Royal Statistical Society: Series C: Applied Statistics*, pages 77–96, 2016.
- [13] Yu Ryan Yue and Ji Meng Loh. Bayesian nonparametric estimation of pair correlation function for inhomogeneous spatial point processes. *Journal of Nonparametric Statistics*, 25(2):463–474, 2013.

- [14] E. Spodarev. *Stochastic Geometry, Spatial Statistics and Random Files. Métodos Asintóticos*. Springer, New York, 2013.
- [15] Brian D Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.
- [16] L.J.S. Allen. *An Introduction to Stochastic Processes with Applications to Biology*. Pearson/Prentice Hall, 2003.
- [17] Franco Landriscina. *Simulation and learning*. Springer, 2013.
- [18] A Alan B Pritsker. Compilation of definitions of simulation. *Simulation*, 33(2):61–63, 1979.
- [19] Kathleen Rosen. The history of simulation. In *The comprehensive textbook of healthcare simulation*, pages 5–49. Springer, 2013.
- [20] David Goldsman, Richard E Nance, and James R Wilson. A brief history of simulation revisited. In *Proceedings of the 2010 Winter Simulation Conference*, pages 567–574. IEEE, 2010.
- [21] Felipe Jones, Carlos Eduardo Passos-Neto, and Odonne Freitas Melro Braghiroli. Simulation in medical education: Brief history and methodology. *Principles and Practice of Clinical Research*, 1(2), 2015.
- [22] Anthony J Faria. The changing nature of simulation research: A brief absel history. In *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*, volume 27, 2000.
- [23] Jason P Davis, Kathleen M Eisenhardt, and Christopher B Bingham. Developing theory through simulation methods. *Academy of Management Review*, 32(2):480–499, 2007.
- [24] Dieter W Heermann. Computer-simulation methods. In *Computer Simulation Methods in Theoretical Physics*, pages 8–12. Springer, 1990.
- [25] Francis Neelamkavil. *Computer simulation and modelling*. John Wiley & Sons, Inc., 1987.
- [26] Harvey Gould, Jan Tobochnik, and Wolfgang Christian. An introduction to computer simulation methods. *Comput. Phys*, 10:652–653, 2007.
- [27] Pierre l’Ecuyer. History of uniform random number generation. In *2017 Winter Simulation Conference (WSC)*, pages 202–230. IEEE, 2017.
- [28] Edward J Dudewicz. Random numbers: the need, the history, the generators. In *A Modern Course on Statistical Distributions in Scientific Work*, pages 25–36. Springer, 1975.
- [29] James E Gentle. *Random number generation and Monte Carlo methods*, volume 381. Springer, 2003.
- [30] Christos E Papadopoulos and Hoi Yeung. Uncertainty estimation and monte carlo simulation method. *Flow Measurement and Instrumentation*, 12(4):291–298, 2001.
- [31] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media, 2005.
- [32] Søren Asmussen and Peter W Glynn. *Stochastic simulation: algorithms and analysis*, volume 57. Springer Science & Business Media, 2007.

-
- [33] Carl Graham and Denis Talay. *Stochastic simulation and Monte Carlo methods: mathematical foundations of stochastic simulation*, volume 68. Springer Science & Business Media, 2013.
- [34] Adrian Baddeley, Ege Rubak, and Rolf Turner. *Spatial point patterns: methodology and applications with R*. CRC press, 2015.
- [35] Huu Tue Huynh, Issouf Soumare, et al. *Stochastic simulation and applications in finance with MATLAB programs*, volume 633. John Wiley & Sons, 2011.
- [36] Daniel T Gillespie. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.
- [37] Yang Cao, Hong Li, and Linda Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *The journal of chemical physics*, 121(9):4059–4067, 2004.
- [38] Emrah Yenier and Gail M Atkinson. An equivalent point-source model for stochastic simulation of earthquake ground motions in california. *Bulletin of the Seismological Society of America*, 105(3):1435–1455, 2015.
- [39] Peter A Dowd, Chaoshui Xu, Kanti V Mardia, and Robert J Fowell. A comparison of methods for the stochastic simulation of rock fractures. *Mathematical Geology*, 39(7):697–714, 2007.
- [40] Clarence W Richardson. Stochastic simulation of daily precipitation, temperature, and solar radiation. *Water resources research*, 17(1):182–190, 1981.
- [41] John Von Neumann. Various techniques used in connection with random digits. *John von Neumann, Collected Works*, 5:768–770, 1963.
- [42] William Feller. *An introduction to probability theory and its applications, vol 2*. John Wiley & Sons, 2008.
- [43] JH Curtiss. On the distribution of the quotient of two chance variables. *The Annals of Mathematical Statistics*, 12(4):409–421, 1941.
- [44] MP Quine and DF Watson. Radial generation of n-dimensional poisson processes. *Journal of Applied Probability*, 21(3):548–557, 1984.
- [45] Richard E Rossi, Paul W Borth, and Jon J Tollefson. Stochastic simulation for characterizing ecological spatial patterns and appraising risk. *Ecological Applications*, 3(4):719–735, 1993.