

Entregable 3

Amparo Galvez Vilar

2024-11-20

Contents

Tarea: Estudio de la distribución de los árboles	1
Patrón puntual Roble	1
Conteos por quadrats	2
Métodos de distancia	5
Conclusión	10
Patrón puntual Arce	11
Conteos por quadrats	11
Métodos de distancia	15
Conclusión	20
Patrón puntual Secuoya	21
Conteos por quadrats	21
Métodos de distancia	24
Conclusión	29
Patrón puntual Pino	30
Conteos por quadrats	30
Métodos de distancia	33
Conclusión	38

```
source('utilpp.r')
```

Tarea: Estudio de la distribución de los árboles

Explora los patrones puntuales de robles, arces, secuoyas y pinos, cuyos datos están en los ficheros con los mismos nombres. Pueden ser leídos de la misma forma que en los ejemplos anteriores, excepto el de los pinos, que se carga con el comando `source('pinos.r')`.

tambien hay que decir que modelo se ajusta mas segun el tipo de patron que sea

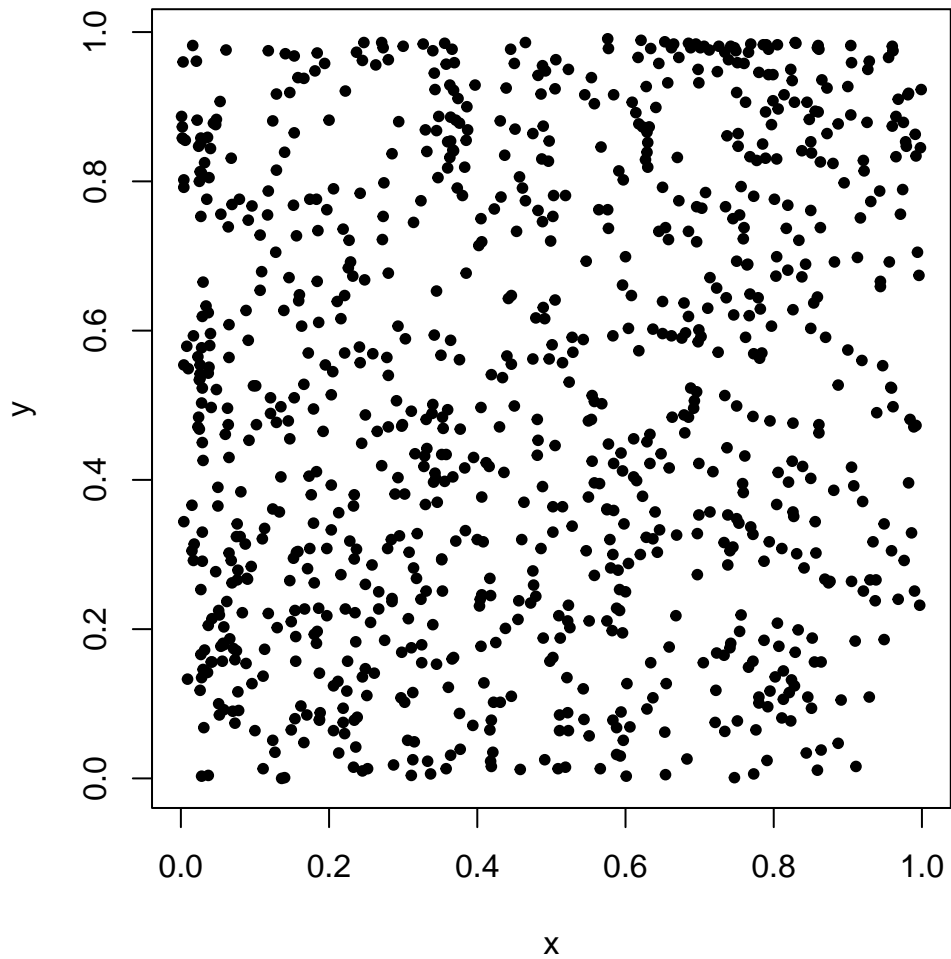
```
# Lectura de los patrones puntuales
robles <- matrix(scan('robles.d'),ncol=2,byrow=T)
arces <- matrix(scan('arces.d'),ncol=2,byrow=T)
secuoyas <- matrix(scan('secuoyas.d'),ncol=2,byrow=T)
pinos <- source('pinos.r')
pinos <- pinos$value
```

Patrón puntual Roble

```
patron <- robles
```

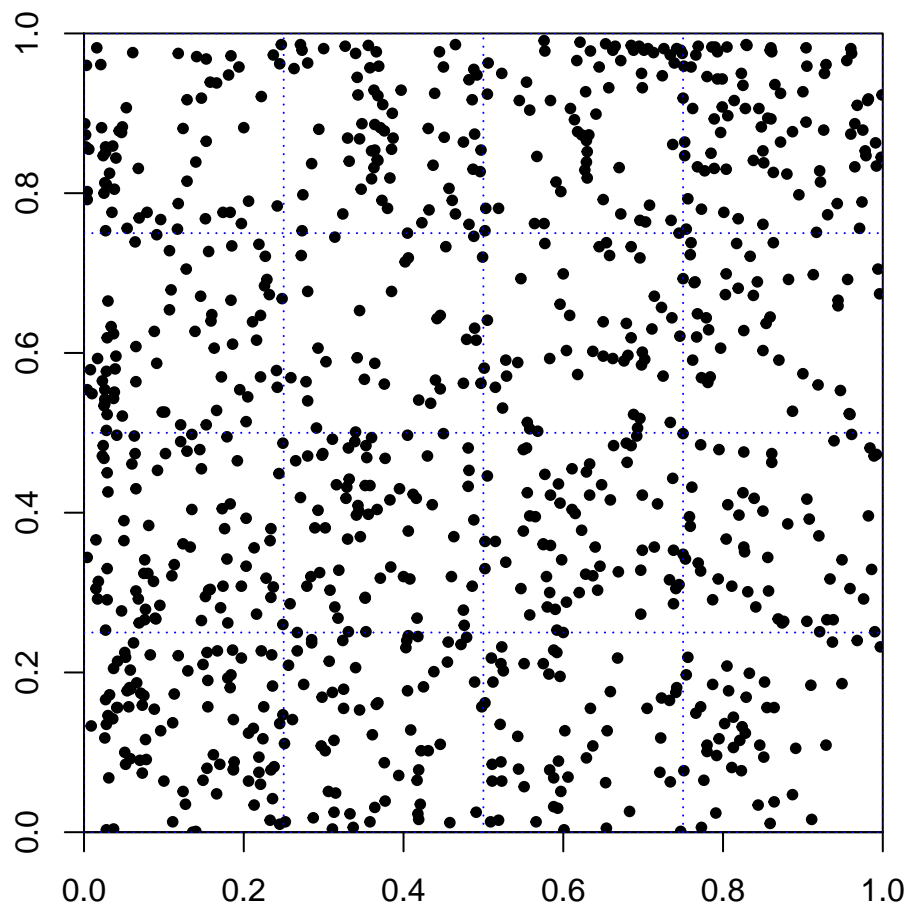
Conteos por quadrats

```
# Representación del patrón puntual en una región cuadrada  
par(pty='s')  
plot(patron,pch=20,xlab='x',ylab='y')
```



```
par(mfrow=c(1,1),pty="s")  
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)  
# La función dibu.qua() dibuja las líneas de un grid de quadrats (4x4)  
dibu.qua(4,4)
```

Análisis mediante un grid de quadrats.



```
# La función cuenta.qua() contabiliza cuantos sucesos hay en cada quadrat
nump.qua<-c(cuenta.qua(patron,4,4))
# El test chi-cuadrado de bondad de ajuste con los conteos observados y con todos los esperados iguales
chisq.test(nump.qua)
```

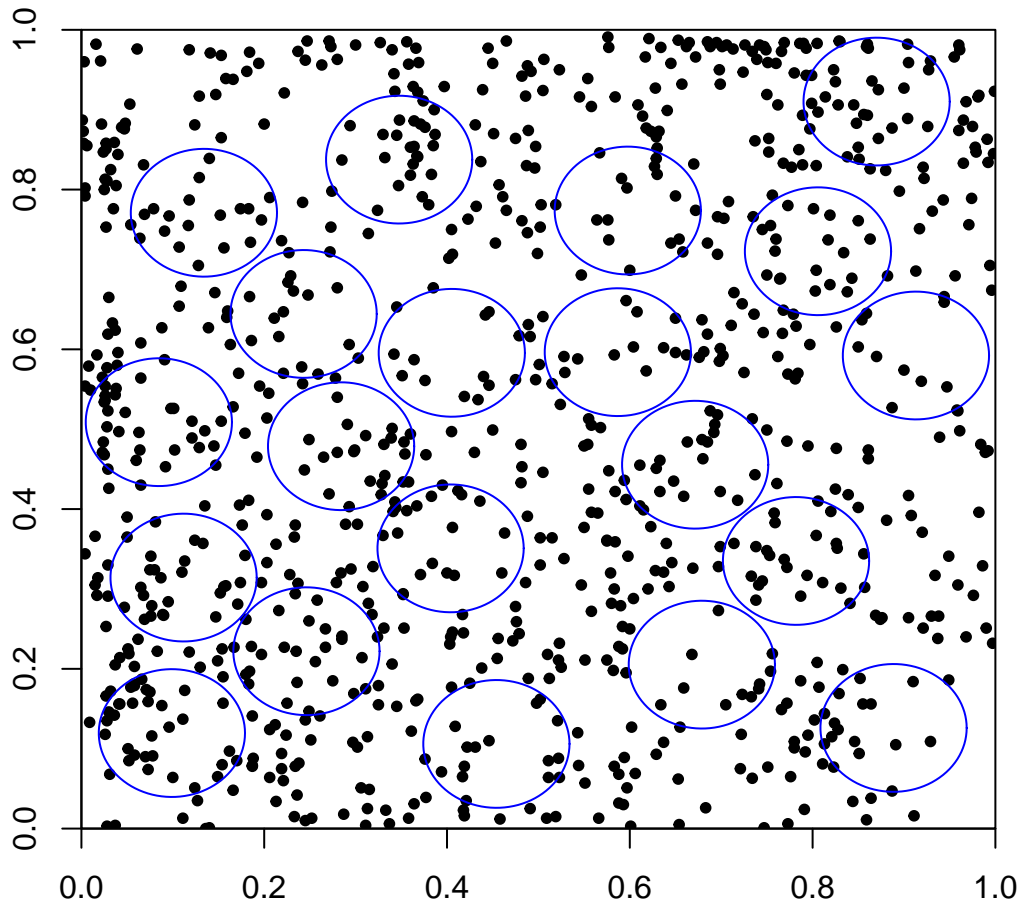
```
##
## Chi-squared test for given probabilities
##
## data:  nump.qua
## X-squared = 52.069, df = 15, p-value = 5.502e-06
```

Dado que el p-valor es muy pequeño, rechazamos la hipótesis de que este patrón sea completamente aleatorio. Observamos que el patrón es agrupado, ya que lo esperado y lo observado son significativamente diferentes.

```
# La función Simqalea() simula los centros de 20 quadrats de radio 0.08
centros.qua<-Simqalea(20,0.08)
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)
```

```
# La función dibalea.qua() dibuja los quadrats circulares simulados
dibalea.qua(centros.qua)
```

Análisis mediante un muestreo con quadrats aleatorios.



```
# La función cupenl.qua() calcula cuantos sucesos hay en cada quadrat
nump.qua<-cupenl.qua(patron,centros.qua)
# La función indices.qua() calcula los índices de agrupación
indices.qua(nump.qua)
```

```
## Índices de agrupamiento:
## -----
##      Varianza Relativa:      2.080774
##      Índice de David-Moore:   1.080774
##      Desigualdad de Lloyd:    1.055567
##      Índice de Morisita:      0.05264622
```

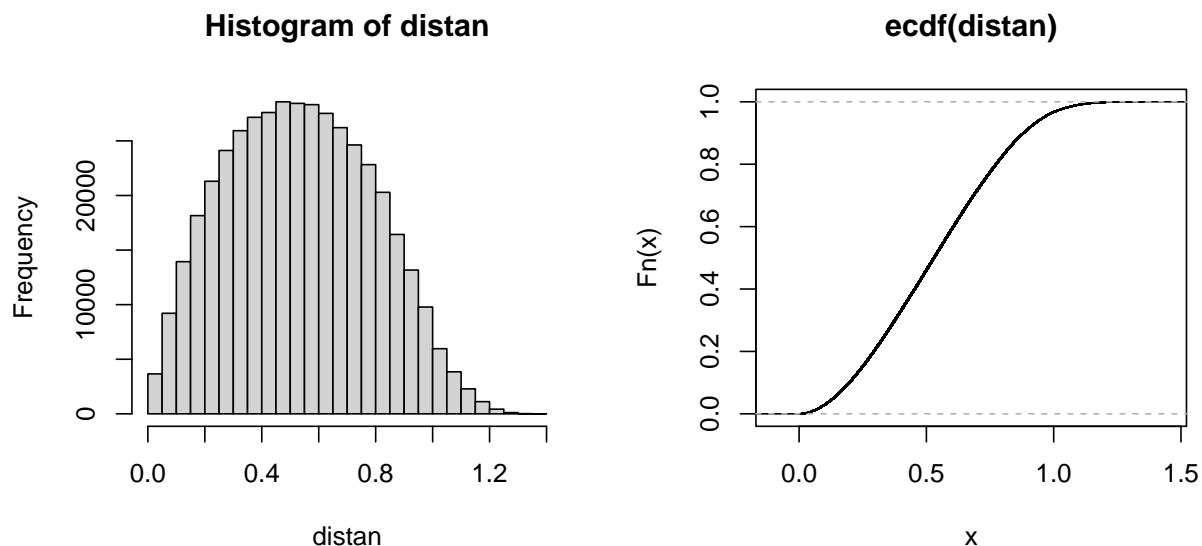
El análisis de los cuadrantes refuerza la interpretación de un patrón agrupado. La varianza relativa (2.162419)

es considerablemente superior a la media, lo que refleja una alta variabilidad en la distribución de los puntos. Los índices calculados también sustentan esta conclusión: el índice de David-Moore (1.162419) y la desigualdad de Lloyd (1.066614) sugieren una moderada concentración de los puntos en ciertas áreas. Por su parte, el índice de Morisita (0.05317327) confirma que la distribución se aleja tanto de la uniformidad como de la aleatoriedad, destacando un nivel de agrupamiento significativo.

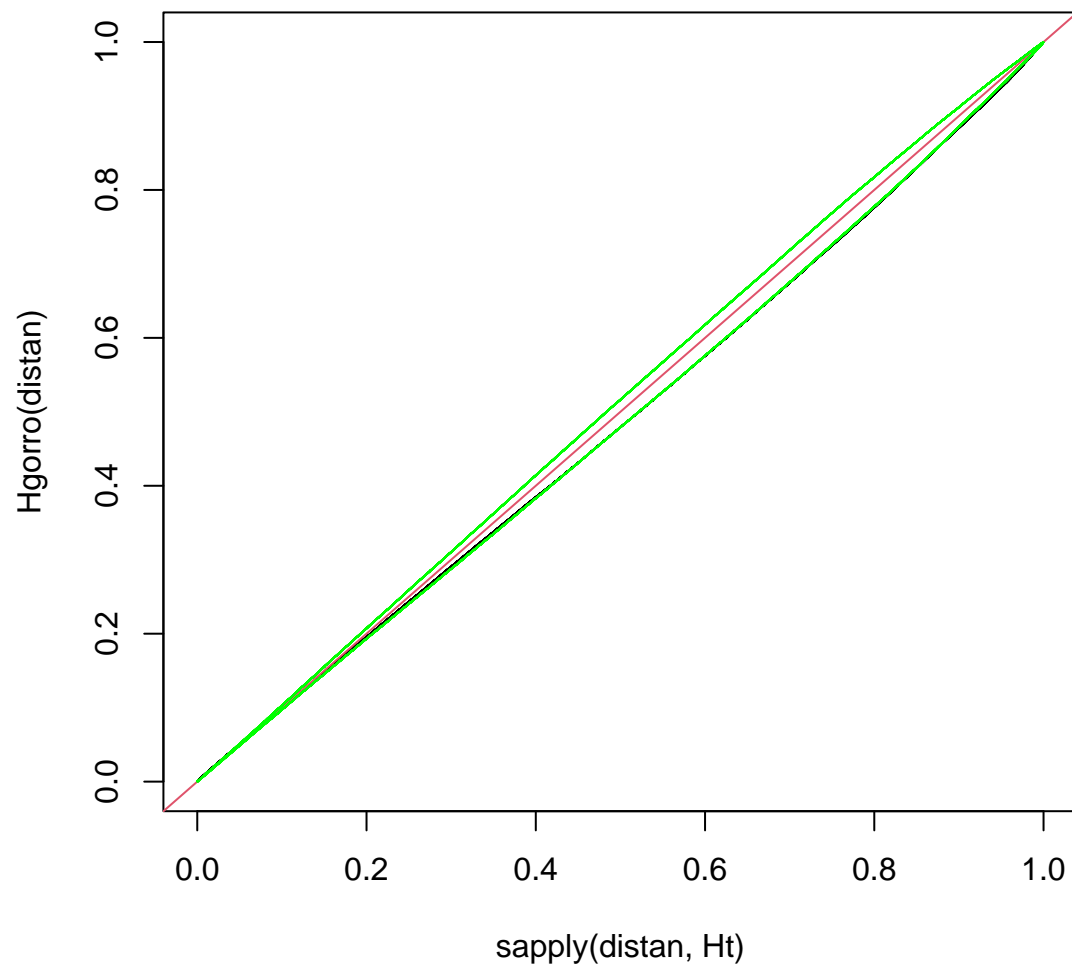
Métodos de distancia

```
# Calculamos todas las distancias entre los sucesos
# La función genérica dist() crea una matriz (n,n) de distancias
# La función sort() ordena y convierte en vector las n*(n-1)/2 distancias
distan<-sort(dist(patron))
par(mfrow=c(1,2))
hist(distan)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Hgorro <- ecdf(distan)
plot(Hgorro, verticals= TRUE, do.p = F)
```

Distancias entre sucesos



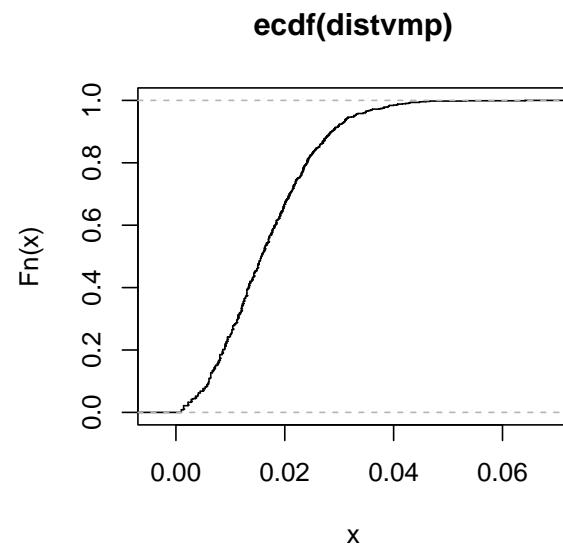
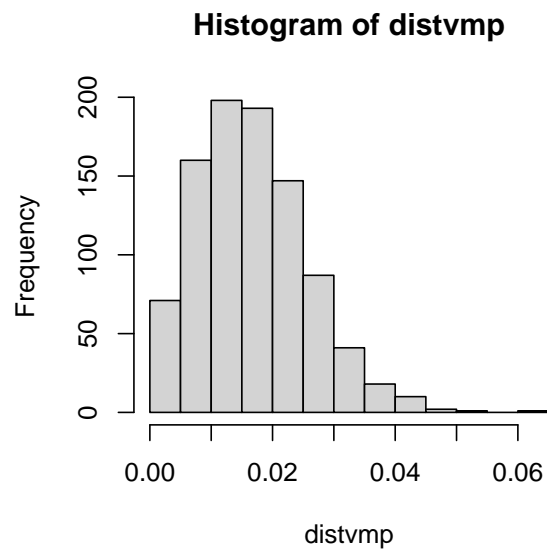
```
# Compararemos la distribución empírica con la teórica Ht()
plot(sapply(distan,Ht),Hgorro(distan),type="l")
# La línea roja marca el ajuste perfecto con la aleatoriedad espacial completa
abline(0,1,col=2)
# La función Henvl() calcula 19 simulaciones de patrones completamente aleatorios
# y determina el mínimo y el máximo de la función H empírica para cada distancia
lims<-Henvl(19,patron)
# Se pintan en verde los límites inferior y superior de las simulaciones
lines(sapply(lims$xl,Ht),Hgorro(distan),col="green")
lines(sapply(lims$xu,Ht),Hgorro(distan),col="green")
```



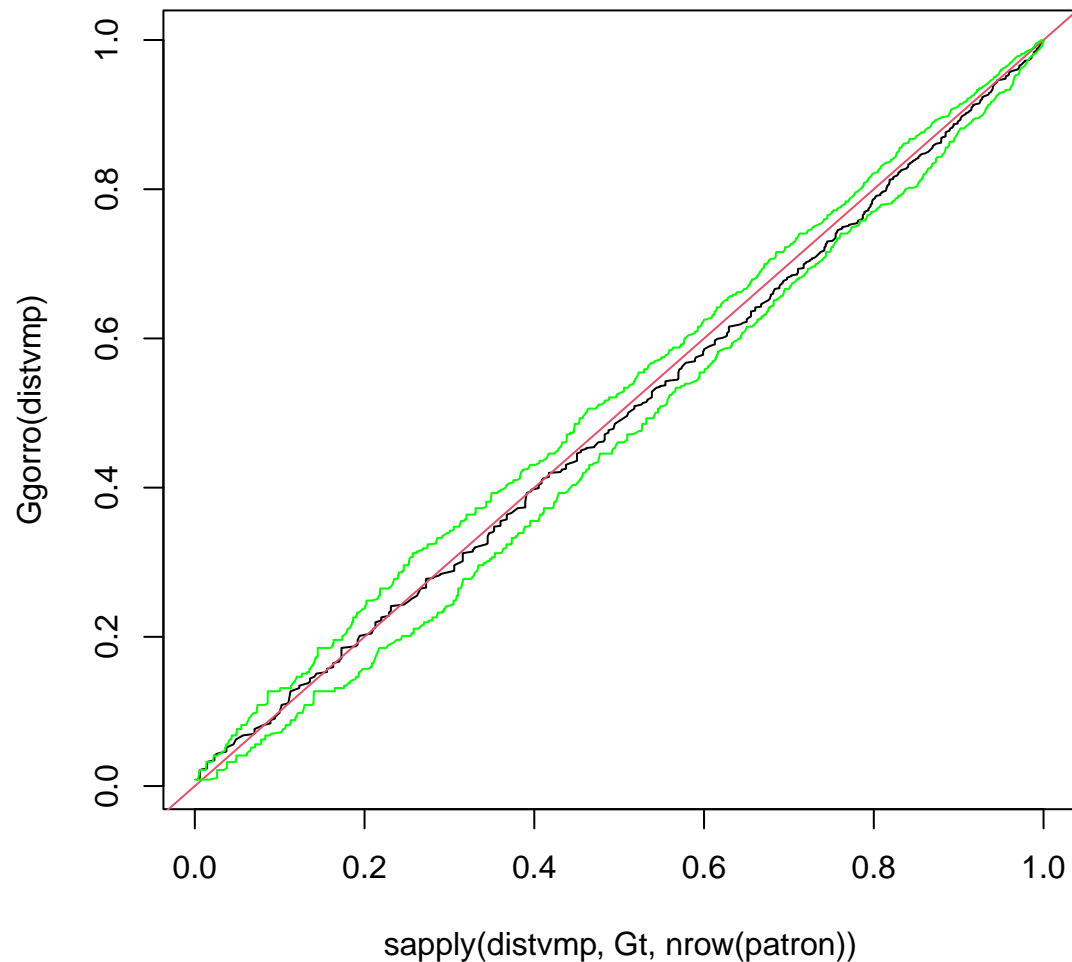
La función H no arroja resultados concluyentes, ya que la línea empírica solo excede mínimamente los límites simulados por debajo. Esto indica que no hay una estructura de agrupamiento o regularidad fuertemente marcada, pero tampoco contradice la tendencia general al agrupamiento identificada por otros métodos.

```
# La función calcdvmp() calcula las distancias al vecino más próximo
distvmp<-sort(calcdvmp(patron))
par(mfrow=c(1,2))
hist(distvmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Ggorro <- ecdf(distvmp)
plot(Ggorro, verticals= TRUE, do.p = F)
```

Distancias al vecino más próximo



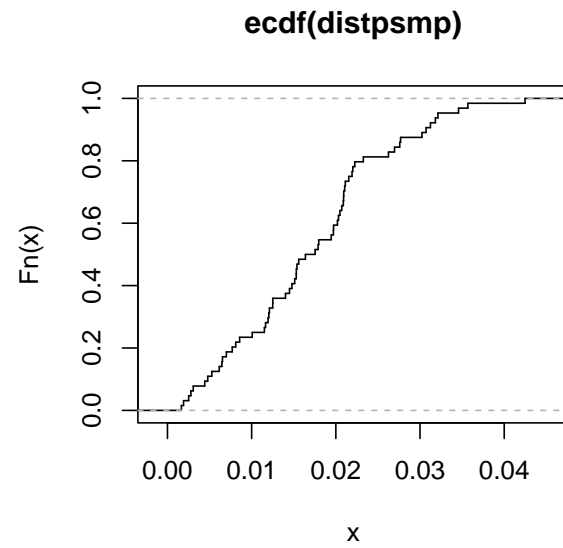
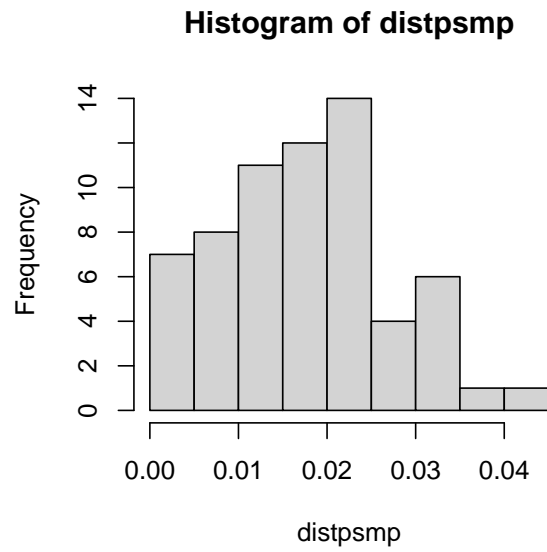
```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distvmp,Gt,nrow(patron)),Ggorro(distvmp),type="l")
abline(0,1,col=2)
lims<-Genvl(19,patron)
lines(sapply(lims$xl,Gt,nrow(patron)),Ggorro(distvmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Ggorro(distvmp),col="green")
```



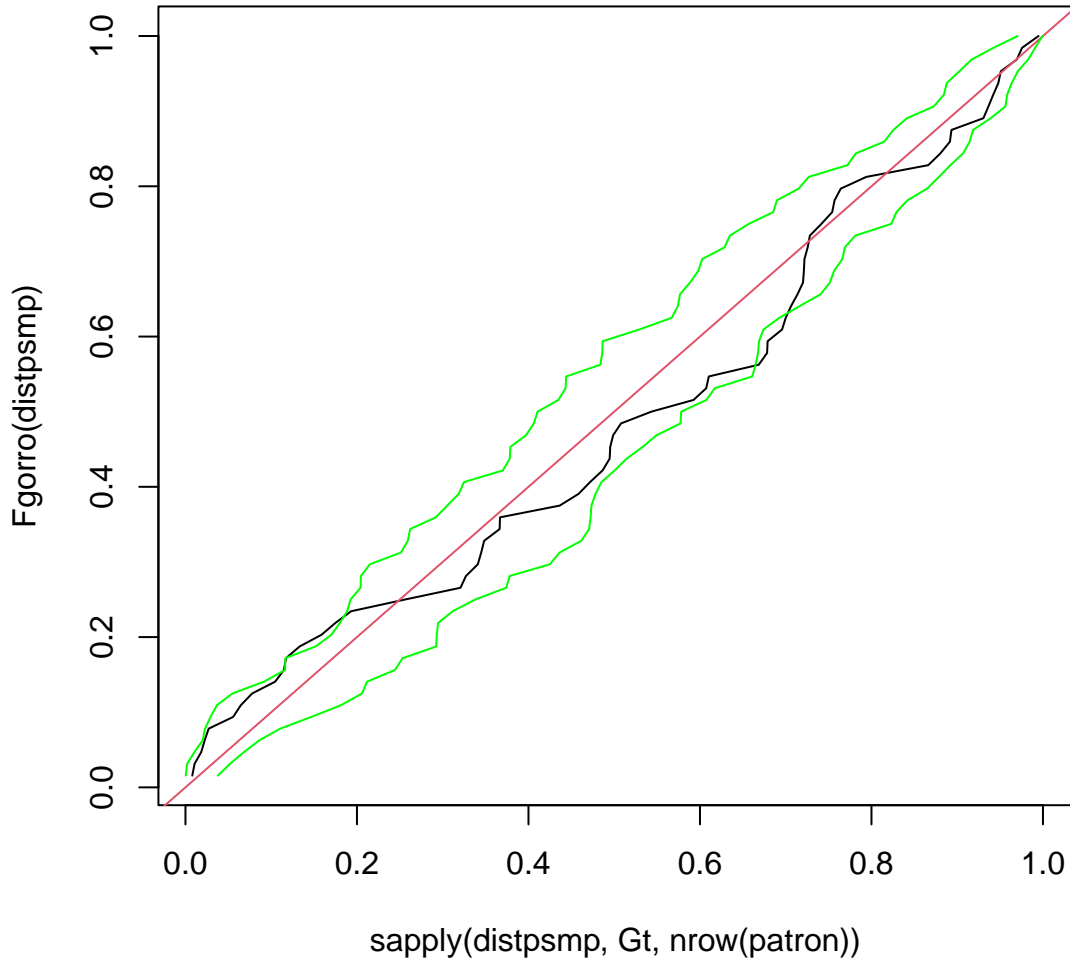
La función G muestra que la línea negra se mantiene cercana a la línea teórica roja, oscilando alrededor de ella sin exceder los límites de las simulaciones. Esto sugiere que las distancias entre los puntos no difieren significativamente de lo esperado bajo un modelo de aleatoriedad, aunque no descarta la presencia de agrupamientos moderados en ciertas zonas.

```
# La función calcdpsmp() calcula las distancias desde 8*8 puntos aleatorios a los sucesos más próximos
distpsmp<-sort(calcdpsmp(patron,8))
par(mfrow=c(1,2))
hist(distpsmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Fgorro <- ecdf(distpsmp)
plot(Fgorro, verticals= TRUE, do.p = F)
```

Distancias de un punto al suceso más próximo



```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distpsmp,Gt,nrow(patron)),Fgorro(distpsmp),type="l")
abline(0,1,col=2)
lims<-Fenvl(19,patron,8)
lines(sapply(lims$xl,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
```



La función F revela que, aunque la línea empírica no excede los límites simulados de manera consistente, en algunos puntos específicos lo hace por debajo. Esto puede indicar áreas del espacio con menor densidad de puntos, pero no lo suficientemente pronunciadas como para alterar significativamente la percepción de agrupamiento.

Conclusión

El análisis realizado con diversos métodos confirma que el patrón observado es agrupado, con ciertas áreas que muestran una concentración de puntos superior a la que sería esperable bajo un modelo aleatorio. Los resultados obtenidos del análisis con grids y cuadrantes indican una alta variabilidad en la distribución de los puntos, lo que sugiere que los puntos no se distribuyen uniformemente, sino que tienden a agruparse en algunas zonas del espacio. La varianza relativa elevada y los índices de concentración, como el de David-Moore y la desigualdad de Lloyd, refuerzan la idea de que los puntos están organizados en grupos moderados, con concentraciones mayores en ciertos sectores del espacio.

Sin embargo, los análisis basados en distancias muestran un comportamiento algo más ambiguo. La función H no presenta un patrón claro de agrupamiento, más allá de una pequeña desviación por debajo de las simulaciones en distancias más cortas, lo que sugiere que los puntos no están tan estrechamente agrupados en

distancias muy pequeñas. La función G, por su parte, muestra que las distancias entre los puntos no exceden las simulaciones teóricas, lo que implica que los puntos no se agrupan fuertemente, pero tampoco se dispersan completamente. Finalmente, la función F muestra que, aunque la línea empírica no supera consistentemente los límites de las simulaciones, en algunos casos sí lo hace ligeramente por debajo, lo que podría indicar una menor densidad de puntos en ciertas áreas, pero sin representar una evidencia fuerte de regularidad.

En conjunto, los resultados apuntan a un patrón agrupado moderadamente. No es un agrupamiento tan marcado como para descartar la posibilidad de aleatoriedad en su totalidad, pero sí se observa una tendencia clara hacia la concentración de puntos en algunas zonas, lo que sugiere una mayor complejidad en la distribución de los puntos que la que un proceso aleatorio podría explicar completamente.

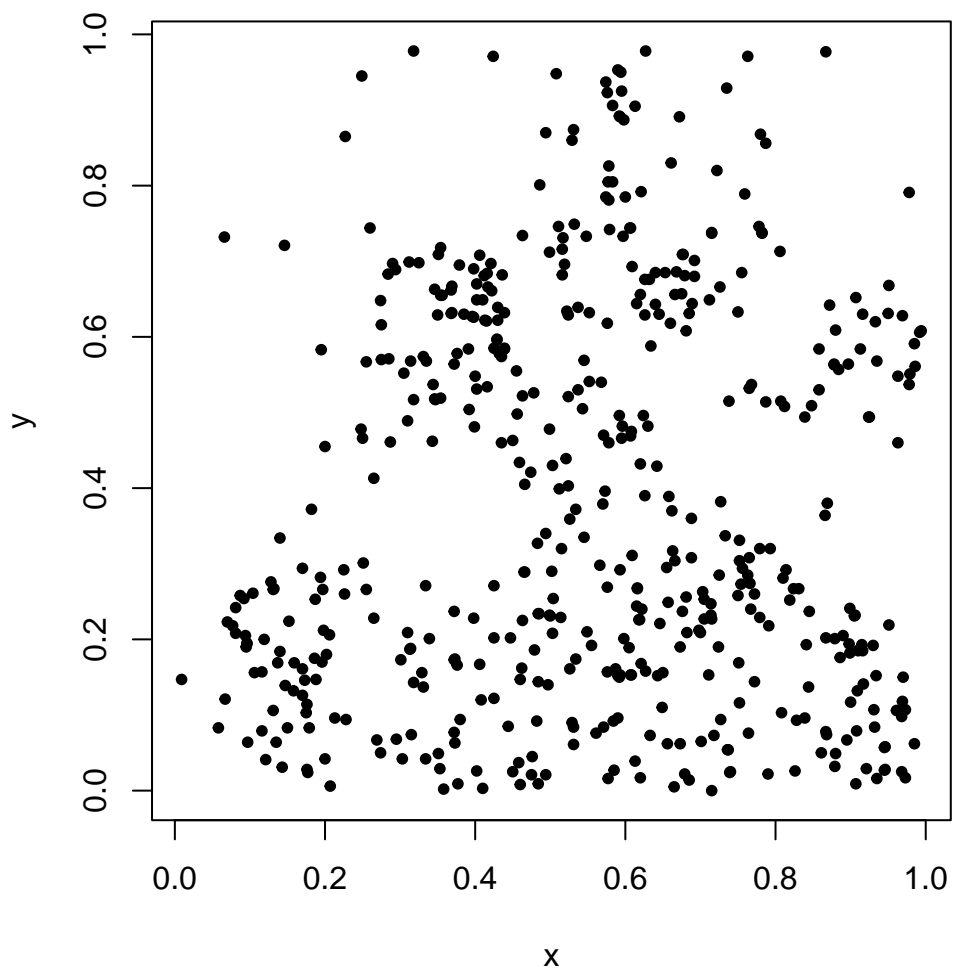
El modelo más adecuado para representar este patrón sería un proceso de agrupación de Poisson, que es capaz de describir este tipo de agrupamiento moderado sin requerir una influencia externa o espacial específica. Sin embargo, si existieran factores espaciales subyacentes que afectaran la distribución de los puntos (como variaciones en la intensidad de eventos en el espacio), un proceso de Cox podría ser una opción alternativa para modelar estos agrupamientos, considerando la posibilidad de que haya variaciones espaciales o ambientales que influyan en la distribución de los puntos.

Patrón puntual Arce

```
patron <- arces
```

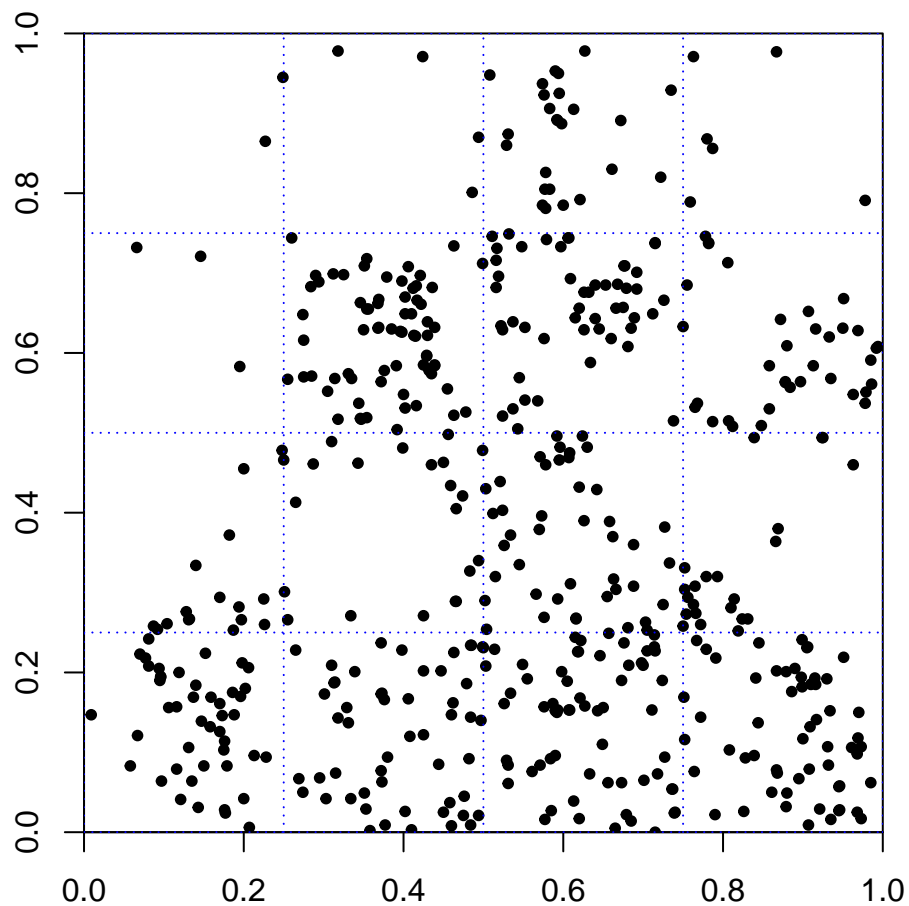
Conteos por quadrats

```
# Representación del patrón puntual en una región cuadrada
par(pty='s')
plot(patron,pch=20,xlab='x',ylab='y')
```



```
par(mfrow=c(1,1),pty="s")
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)
# La función dibu.qua() dibuja las líneas de un grid de quadrats (4x4)
dibu.qua(4,4)
```

Análisis mediante un grid de quadrats.



```
# La función cuenta.qua() contabiliza cuantos sucesos hay en cada quadrat
nump.qua<-c(cuenta.qua(patron,4,4))
# El test chi-cuadrado de bondad de ajuste con los conteos observados y con todos los esperados iguales
chisq.test(nump.qua)
```

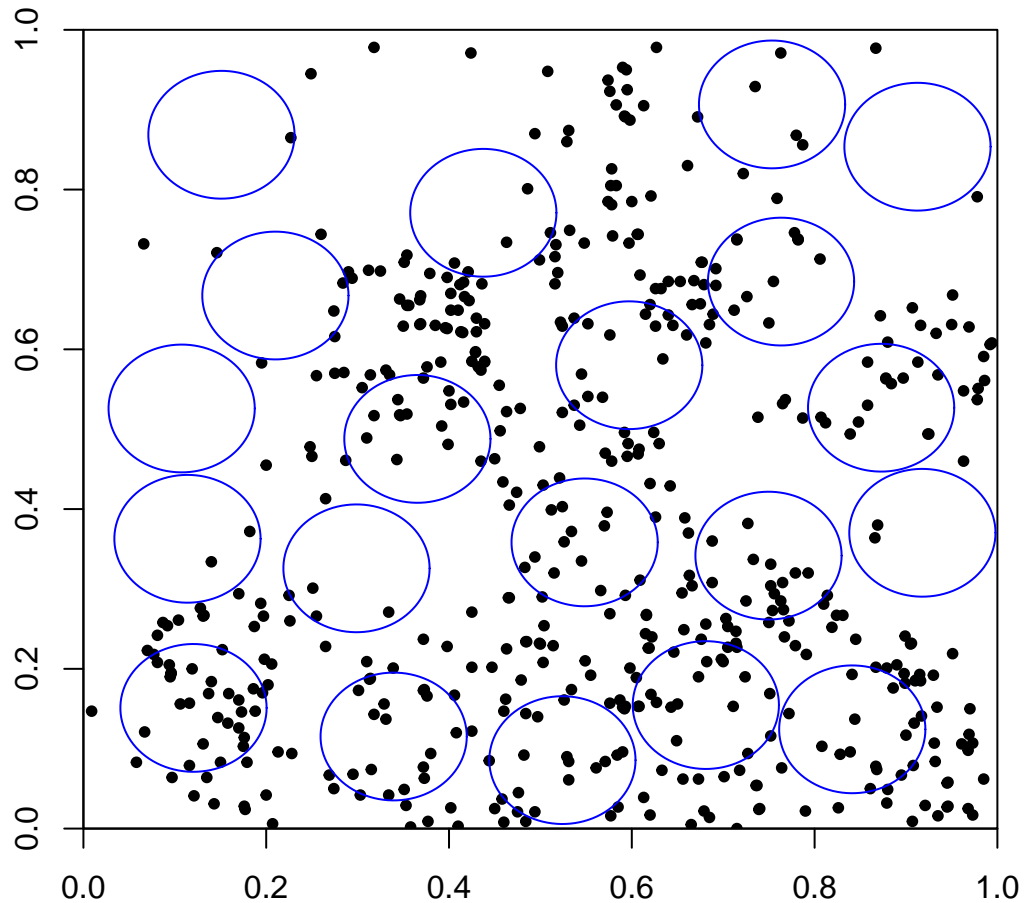
```
##
## Chi-squared test for given probabilities
##
## data:  nump.qua
## X-squared = 241.56, df = 15, p-value < 2.2e-16
```

Dado que el p-valor es muy pequeño, rechazamos la hipótesis de que este patrón sea completamente aleatorio. Observamos que el patrón es agrupado, ya que lo esperado y lo observado son significativamente diferentes.

```
# La función Simqalea() simula los centros de 20 quadrats de radio 0.08
centros.qua<-Simqalea(20,0.08)
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)
```

```
# La función dibalea.qua() dibuja los quadrats circulares simulados
dibalea.qua(centros.qua)
```

Análisis mediante un muestreo con quadrats aleatorios.



```
# La función cupenl.qua() calcula cuantos sucesos hay en cada quadrat
nump.qua<-cupenl.qua(patron,centros.qua)
# La función indices.qua() calcula los índices de agrupación
indices.qua(nump.qua)
```

```
## Índices de agrupamiento:
## -----
##   Varianza Relativa:      6.006719
##   Índice de David-Moore:  5.006719
##   Desigualdad de Lloyd:   1.53263
##   Índice de Morisita:     0.0754352
```

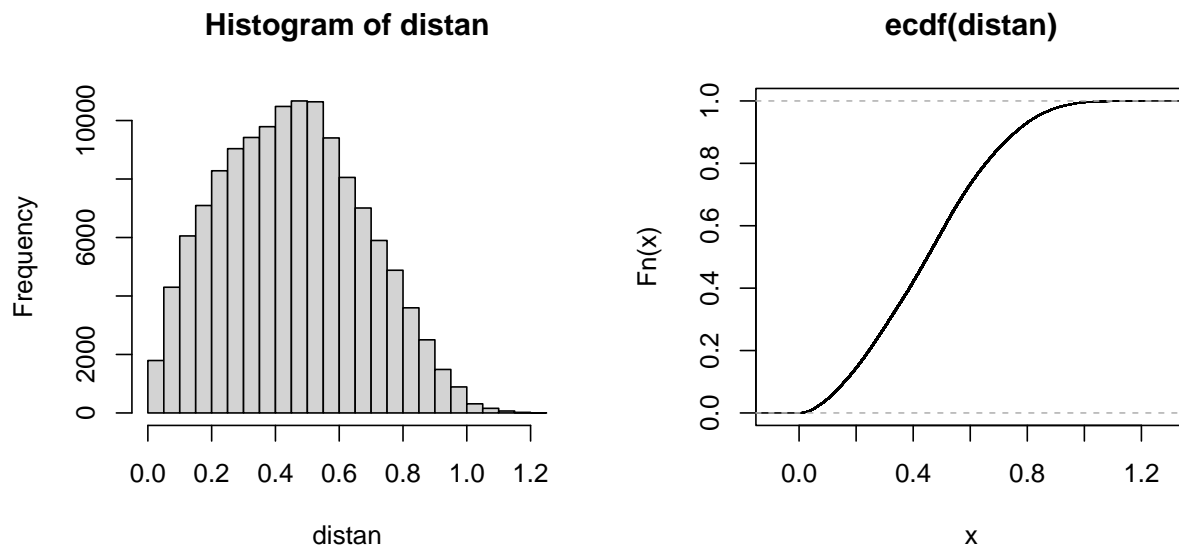
El análisis de los puntos en los cuadrantes revela una alta variabilidad, con una varianza relativa de 7.623518,

muy superior a 1, lo que indica un patrón claramente agrupado. Este agrupamiento es respaldado por el índice de David-Moore (6.623518), que se encuentra muy por encima de cero, y la desigualdad de Lloyd (1.596713), que señala una desigualdad notable en la distribución. El índice de Morisita (0.07847214), aunque sin un valor de referencia fijo, permite comparar este patrón con otros y confirma la tendencia al agrupamiento.

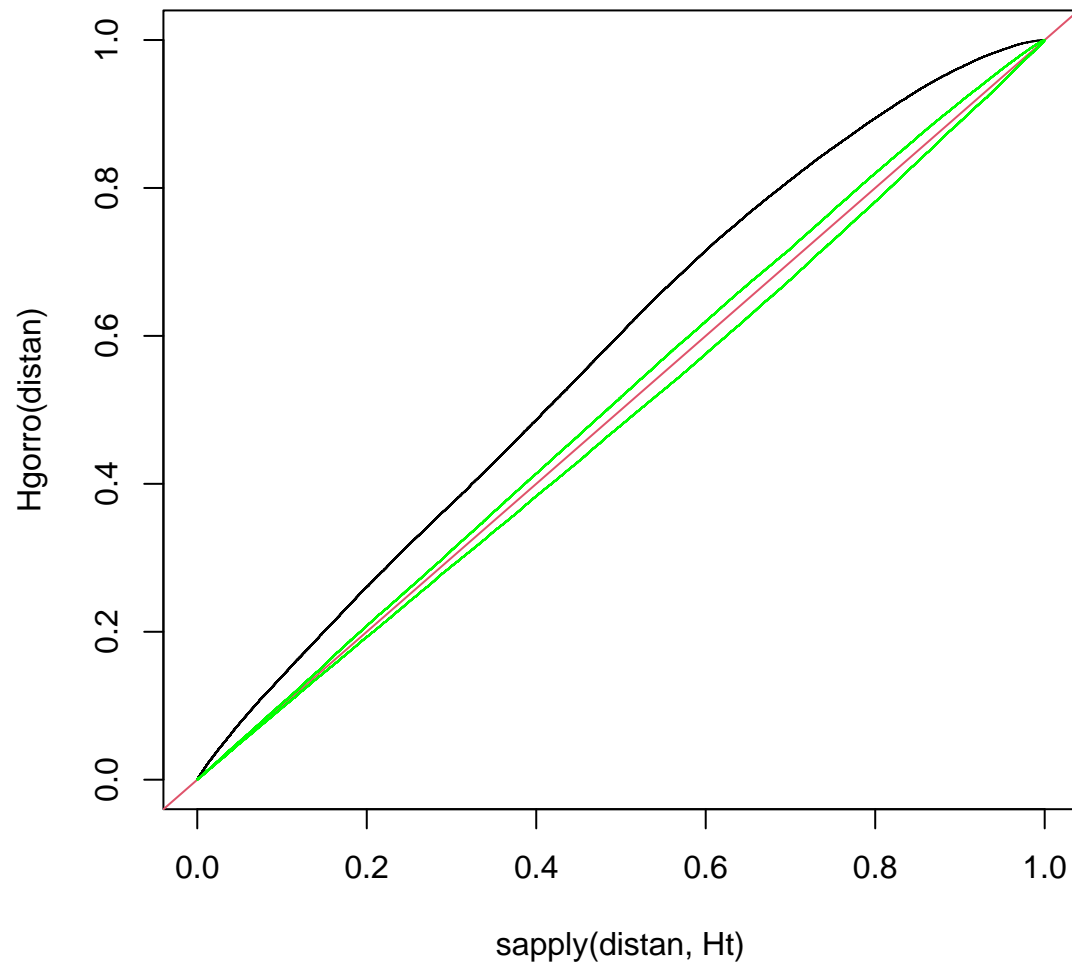
Métodos de distancia

```
# Calculamos todas las distancias entre los sucesos
# La función genérica dist() crea una matriz (n,n) de distancias
# La función sort() ordena y convierte en vector las n*(n-1)/2 distancias
distan<-sort(dist(patron))
par(mfrow=c(1,2))
hist(distan)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Hgorro <- ecdf(distan)
plot(Hgorro, verticals= TRUE, do.p = F)
```

Distancias entre sucesos



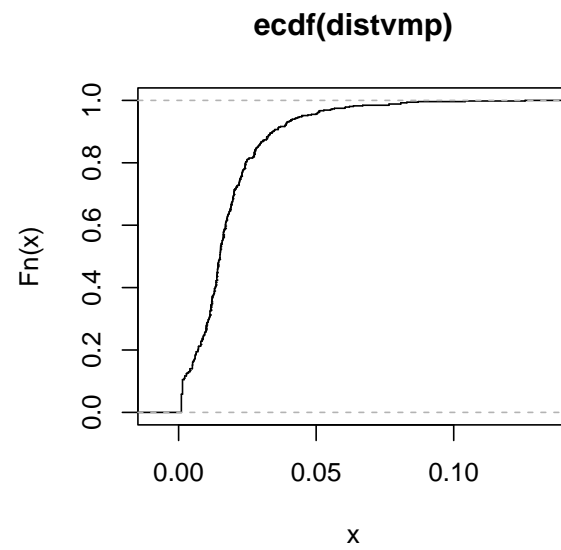
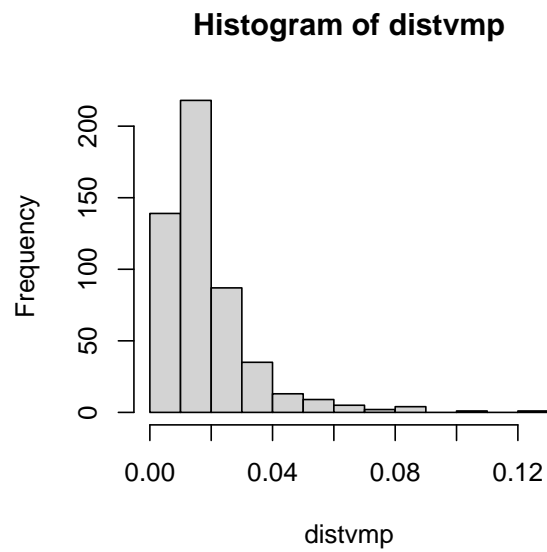
```
# Compararemos la distribución empírica con la teórica Ht()
plot(sapply(distan,Ht),Hgorro(distan),type="l")
# La línea roja marca el ajuste perfecto con la aleatoridad espacial completa
abline(0,1,col=2)
# La función Henvl() calcula 19 simulaciones de patrones completamente aleatorios
# y determina el mínimo y el máximo de la función H empírica para cada distancia
lims<-Henvl(19,patron)
# Se pintan en verde los límites inferior y superior de las simulaciones
lines(sapply(lims$xl,Ht),Hgorro(distan),col="green")
lines(sapply(lims$xu,Ht),Hgorro(distan),col="green")
```



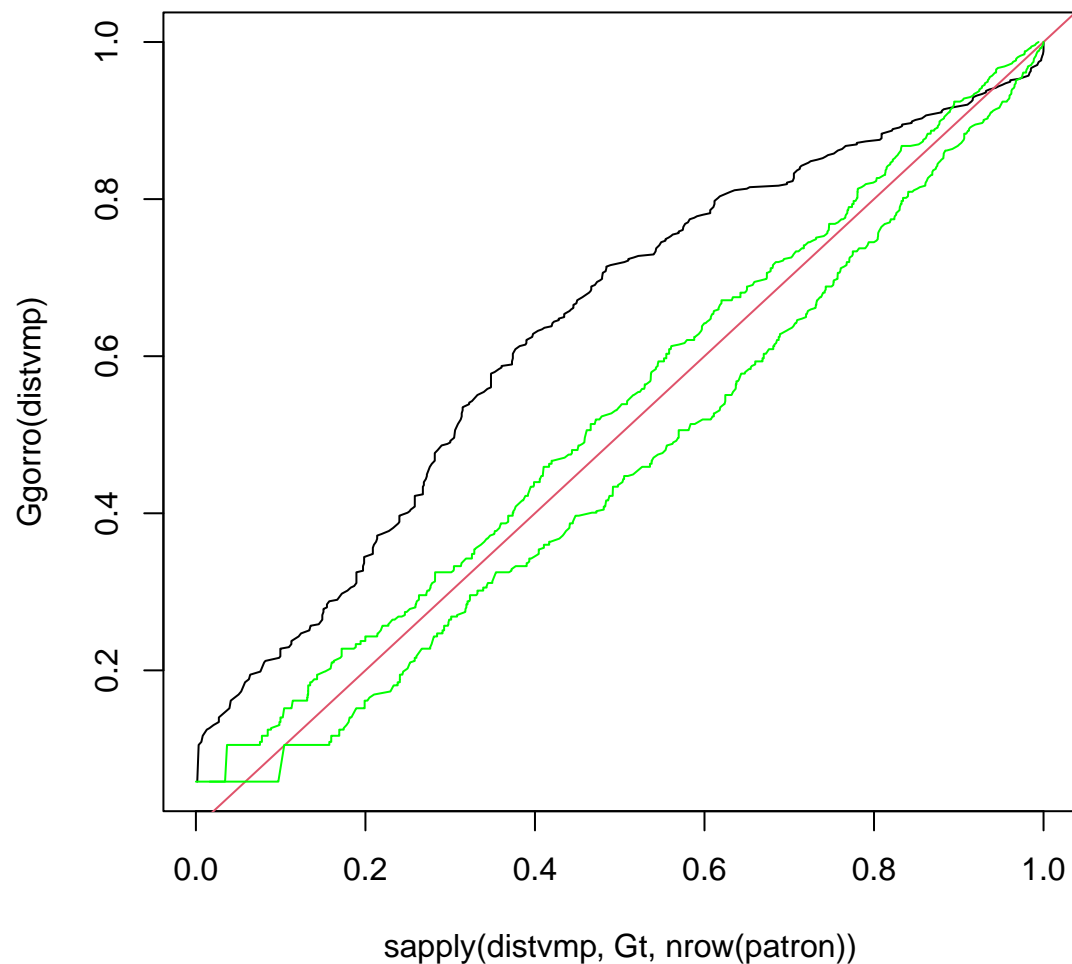
La distribución empírica obtenida en este análisis supera por la parte superior a las simulaciones de referencia, lo que indica una mayor densidad local de puntos en comparación con un patrón aleatorio. Este comportamiento sugiere que los puntos tienden a agruparse, confirmando la presencia de un patrón agrupado.

```
# La función calcdvmp() calcula las distancias al vecino más próximo
distvmp<-sort(calcdvmp(patron))
par(mfrow=c(1,2))
hist(distvmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Ggorro <- ecdf(distvmp)
plot(Ggorro, verticals= TRUE, do.p = F)
```

Distancias al vecino más próximo



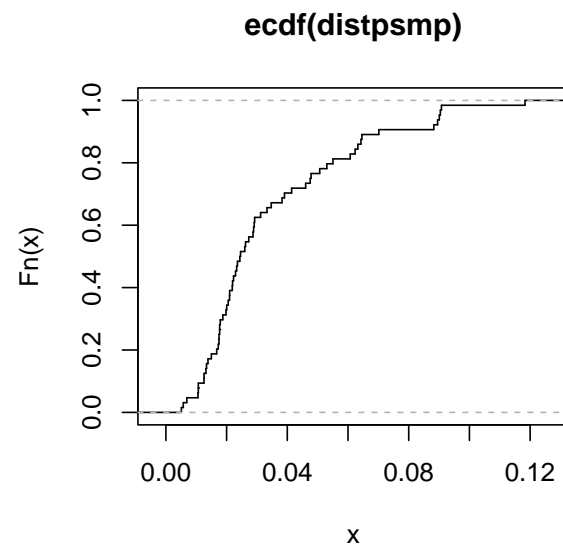
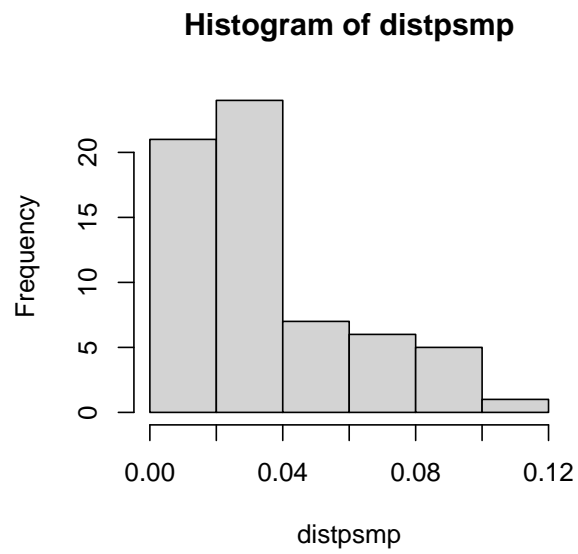
```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distvmp,Gt,nrow(patron)),Ggorro(distvmp),type="l")
abline(0,1,col=2)
lims<-Genvl(19,patron)
lines(sapply(lims$x1,Gt,nrow(patron)),Ggorro(distvmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Ggorro(distvmp),col="green")
```



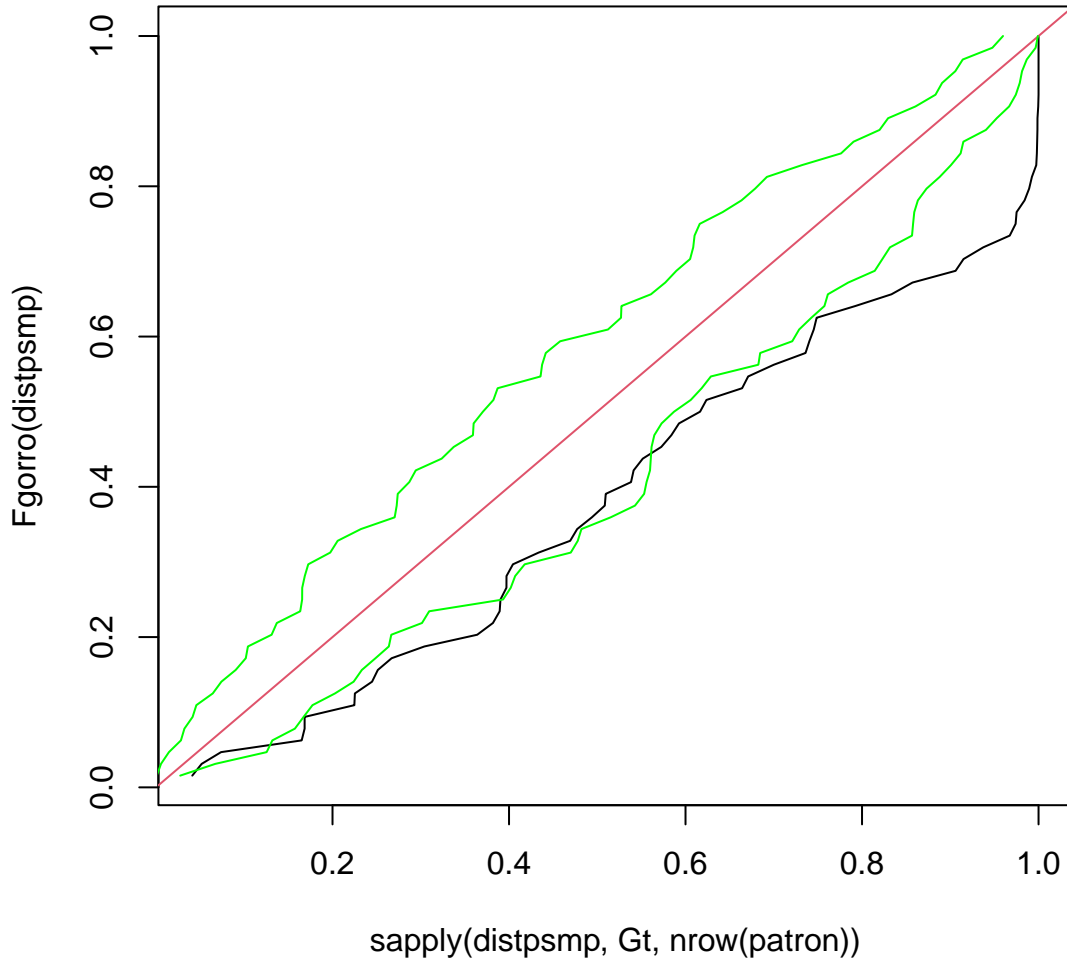
Este método respalda la conclusión anterior. La distribución empírica del patrón excede por encima las simulaciones aleatorias, mostrando que las distancias observadas entre puntos y sus vecinos inmediatos son menores que las esperadas en un proceso aleatorio. Esto refuerza la interpretación de que se trata de un patrón no aleatorio, sino agrupado.

```
# La función calcdpsmp() calcula las distancias desde 8*8 puntos aleatorios a los sucesos más próximos
distpsmp<-sort(calcdpsmp(patron,8))
par(mfrow=c(1,2))
hist(distpsmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Fgorro <- ecdf(distpsmp)
plot(Fgorro, verticals= TRUE, do.p = F)
```

Distancias de un punto al suceso más próximo



```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distpsmp,Gt,nrow(patron)),Fgorro(distpsmp),type="l")
abline(0,1,col=2)
lims<-Fenvl(19,patron,8)
lines(sapply(lims$xl,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
```



Aquí, la distribución empírica se sitúa por debajo de las simulaciones, lo que implica que la función de distribución del patrón crece más lentamente que la de un patrón aleatorio. Este resultado indica que los puntos están más concentrados en ciertas áreas, confirmando nuevamente la existencia de un patrón agrupado.

Conclusión

El análisis exhaustivo realizado a través de diversos métodos confirma de manera consistente que el patrón observado es agrupado y no aleatorio. Los resultados del análisis con grids y muestreo aleatorio de cuadrantes destacan una alta variabilidad en la distribución de los puntos, evidenciada por la varianza relativa elevada y el comportamiento de los indicadores de agrupamiento. Asimismo, los métodos basados en distancias corroboran esta conclusión, mostrando que las distribuciones empíricas difieren significativamente de las simulaciones aleatorias.

En conjunto, los hallazgos reflejan una clara tendencia de los puntos a concentrarse en áreas específicas, rechazando la hipótesis de aleatoriedad y confirmando la existencia de un patrón fuertemente agrupado.

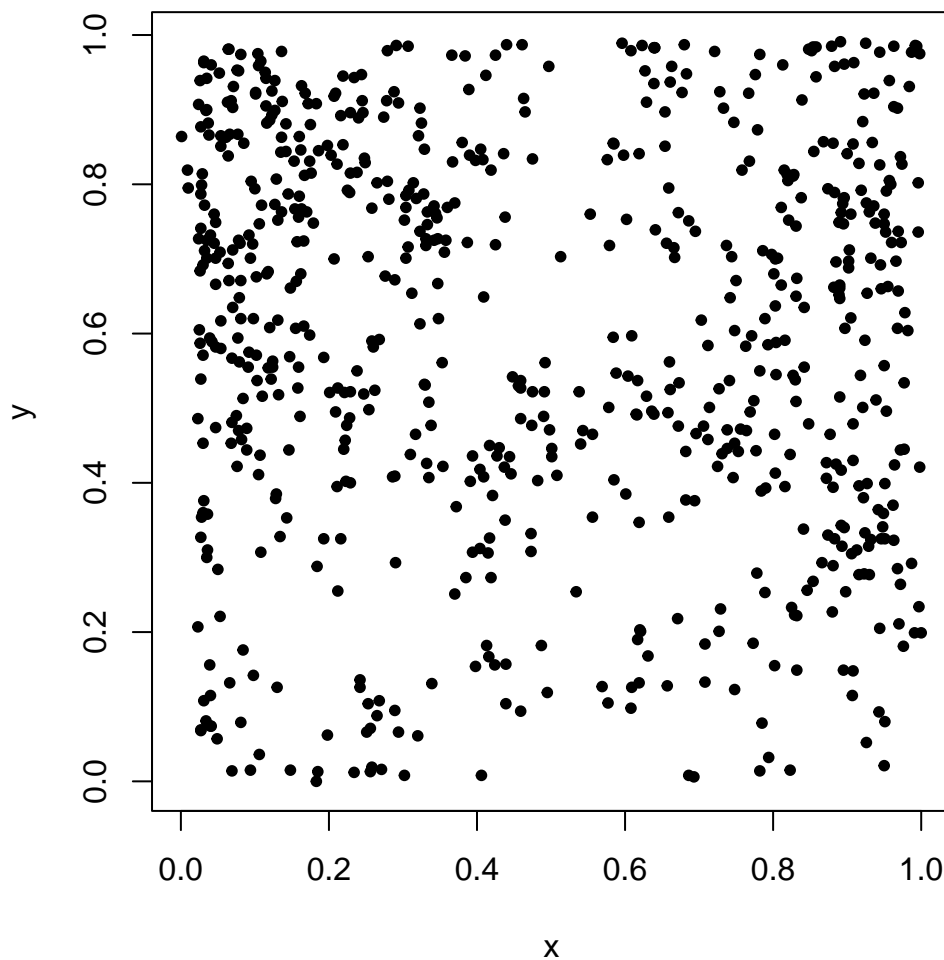
El modelo más adecuado sería un proceso de agrupación de Poisson si los agrupamientos tienen límites definidos, o un proceso de Cox si los agrupamientos están influenciados por variaciones espaciales subyacentes.

Patrón puntual Secuoya

```
patron <- secuoyas
```

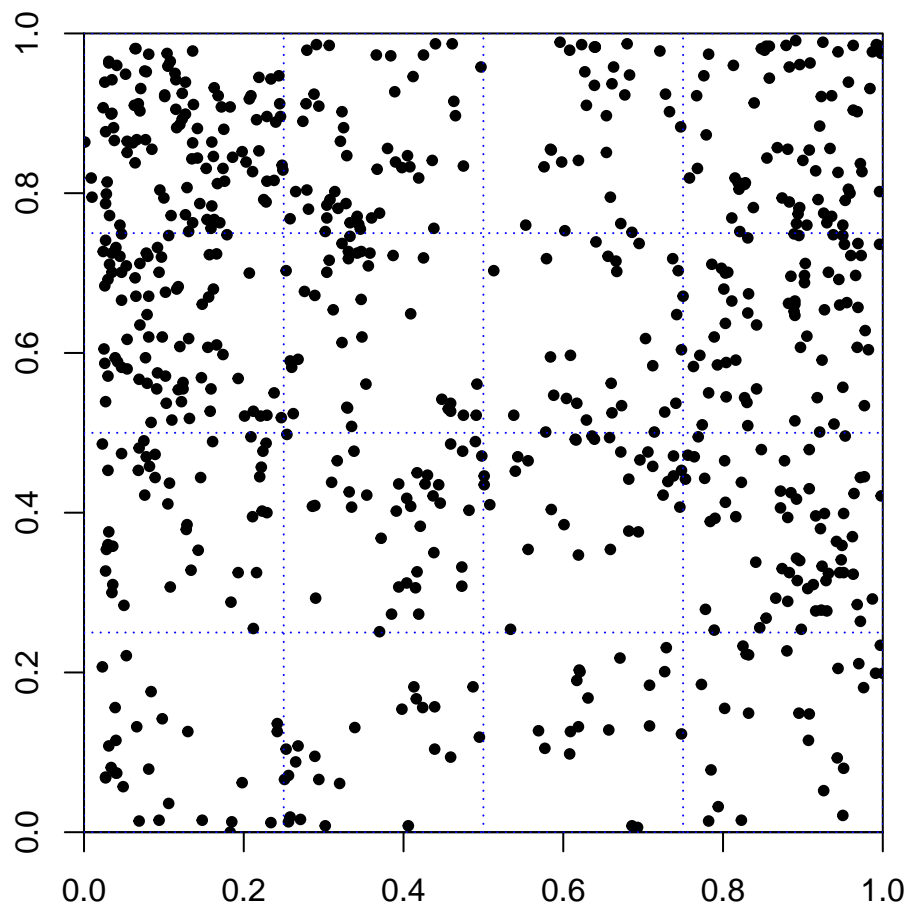
Conteos por quadrats

```
# Representación del patrón puntual en una región cuadrada  
par(pty='s')  
plot(patron,pch=20,xlab='x',ylab='y')
```



```
par(mfrow=c(1,1),pty="s")  
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)  
# La función dibu.qua() dibuja las líneas de un grid de quadrats (4x4)  
dibu.qua(4,4)
```

Análisis mediante un grid de quadrats.



```
# La función cuenta.qua() contabiliza cuantos sucesos hay en cada quadrat
nump.qua<-c(cuenta.qua(patron,4,4))
# El test chi-cuadrado de bondad de ajuste con los conteos observados y con todos los esperados iguales
chisq.test(nump.qua)
```

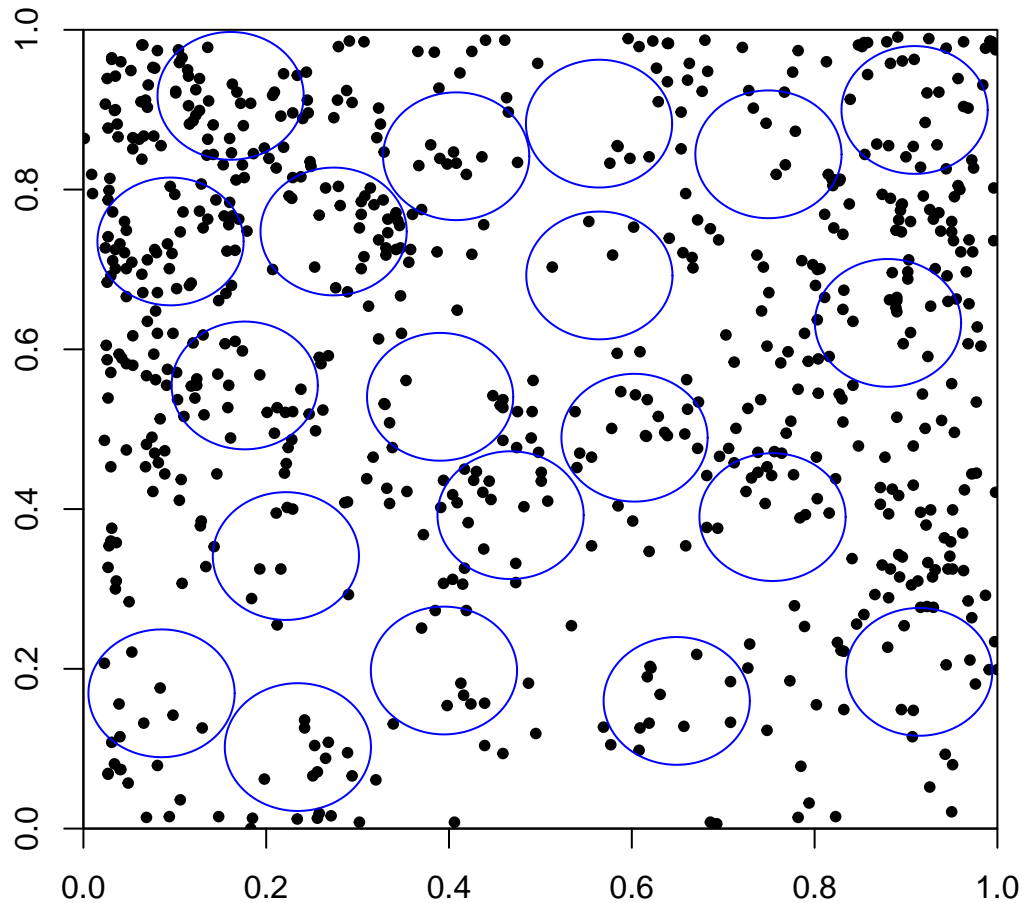
```
##
## Chi-squared test for given probabilities
##
## data:  nump.qua
## X-squared = 181.69, df = 15, p-value < 2.2e-16
```

Dado que el p-valor es muy pequeño, rechazamos la hipótesis de que este patrón sea completamente aleatorio. Observamos que el patrón es agrupado, ya que lo esperado y lo observado son significativamente diferentes.

```
# La función Simqalea() simula los centros de 20 quadrats de radio 0.08
centros.qua<-Simqalea(20,0.08)
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)
```

```
# La función dibalea.qua() dibuja los quadrats circulares simulados
dibalea.qua(centros.qua)
```

Análisis mediante un muestreo con quadrats aleatorios.



```
# La función cupenl.qua() calcula cuantos sucesos hay en cada quadrat
nump.qua<-cupenl.qua(patron,centros.qua)
# La función indices.qua() calcula los índices de agrupación
indices.qua(nump.qua)
```

```
## Índices de agrupamiento:
## -----
##      Varianza Relativa:      6.542927
##      Índice de David-Moore:   5.542927
##      Desigualdad de Lloyd:    1.368301
##      Índice de Morisita:      0.0675526
```

El análisis de los puntos en los cuadrantes revela una alta variabilidad, con una varianza relativa de 4.787587,

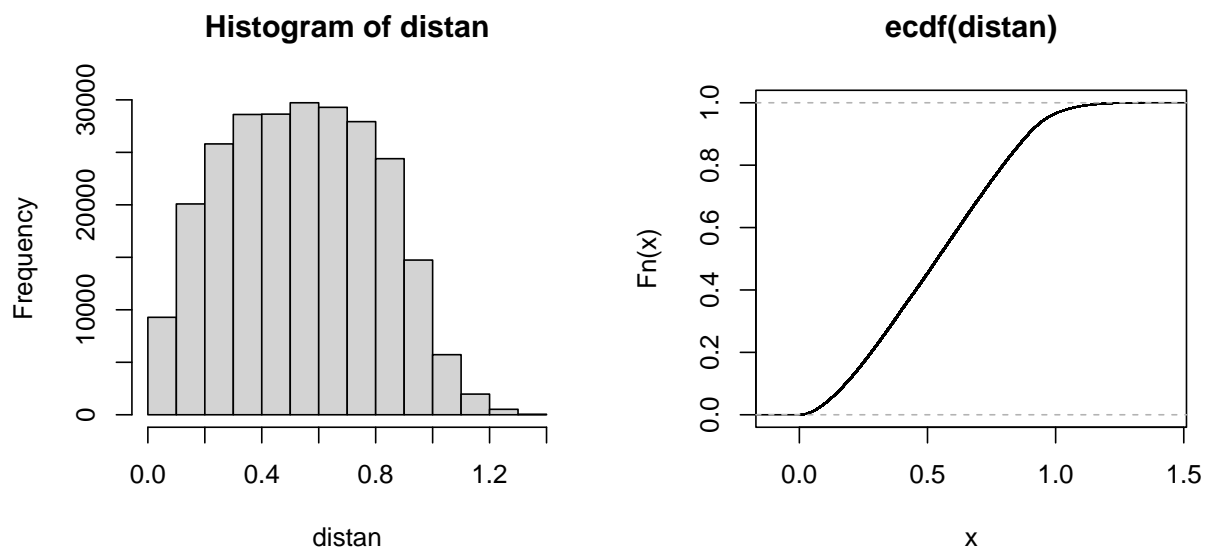
significativamente superior a la media. Esto indica un patrón claramente agrupado.

Los indicadores refuerzan esta conclusión: el índice de David-Moore (3.787587) muestra una fuerte tendencia al agrupamiento, y la desigualdad de Lloyd (1.271512) sugiere una concentración notable. El índice de Morisita (0.06294319), aunque sin valor de referencia absoluto, permite comparar patrones y confirma el carácter altamente agrupado de esta distribución.

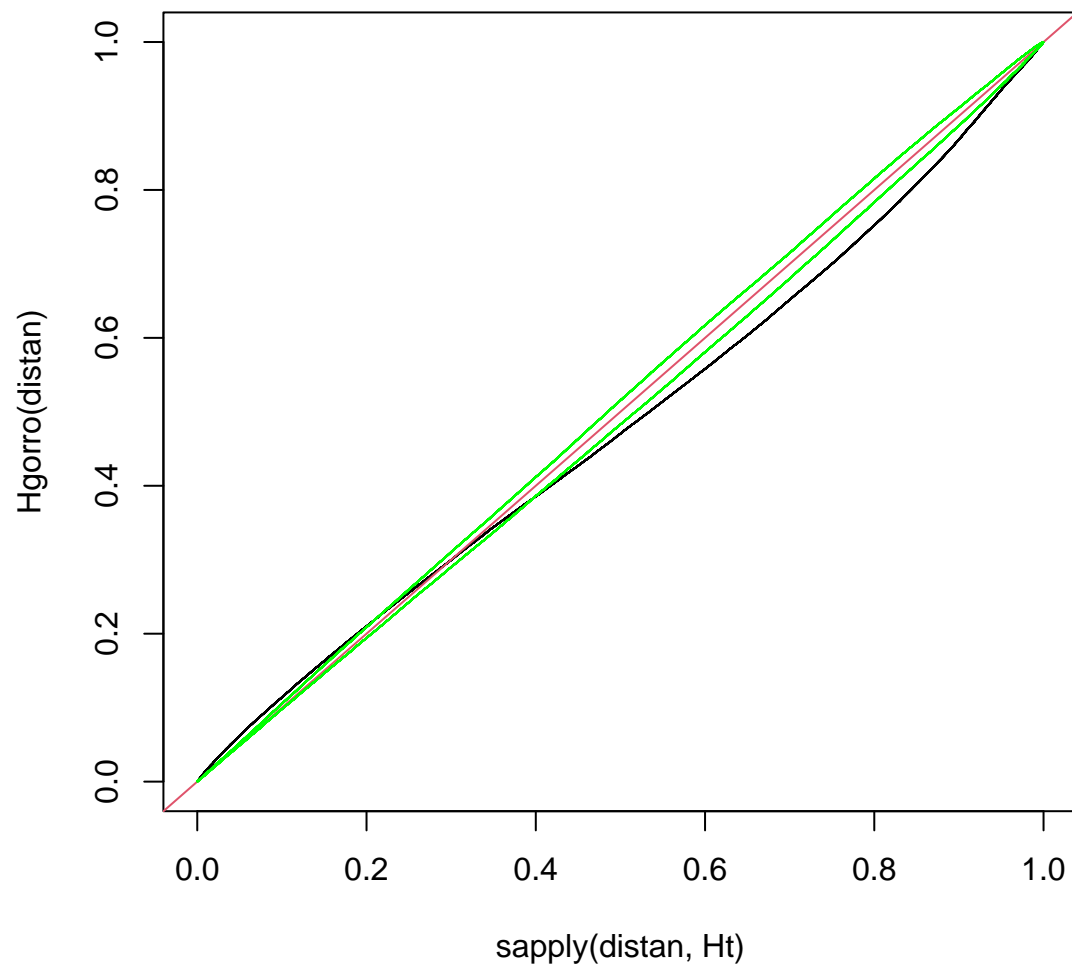
Métodos de distancia

```
# Calculamos todas las distancias entre los sucesos
# La función genérica dist() crea una matriz (n,n) de distancias
# La función sort() ordena y convierte en vector las n*(n-1)/2 distancias
distan<-sort(dist(patron))
par(mfrow=c(1,2))
hist(distan)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Hgorro <- ecdf(distan)
plot(Hgorro, verticals= TRUE, do.p = F)
```

Distancias entre sucesos



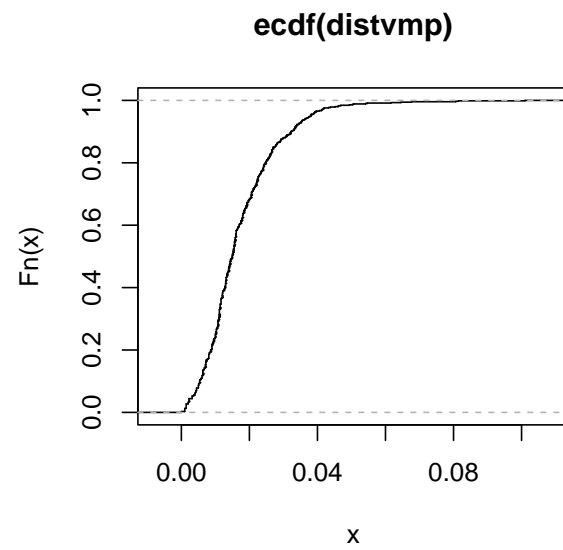
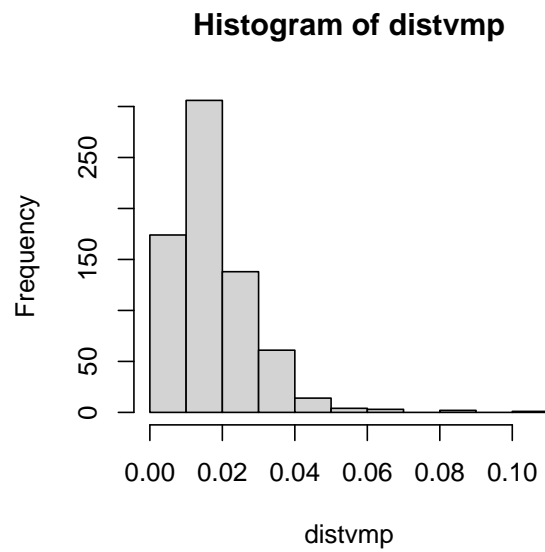
```
# Compararemos la distribución empírica con la teórica Ht()
plot(sapply(distan,Ht),Hgorro(distan),type="l")
# La línea roja marca el ajuste perfecto con la aleatoriedad espacial completa
abline(0,1,col=2)
# La función Henvl() calcula 19 simulaciones de patrones completamente aleatorios
# y determina el mínimo y el máximo de la función H empírica para cada distancia
lims<-Henvl(19,patron)
# Se pintan en verde los límites inferior y superior de las simulaciones
lines(sapply(lims$xl,Ht),Hgorro(distan),col="green")
lines(sapply(lims$xu,Ht),Hgorro(distan),col="green")
```



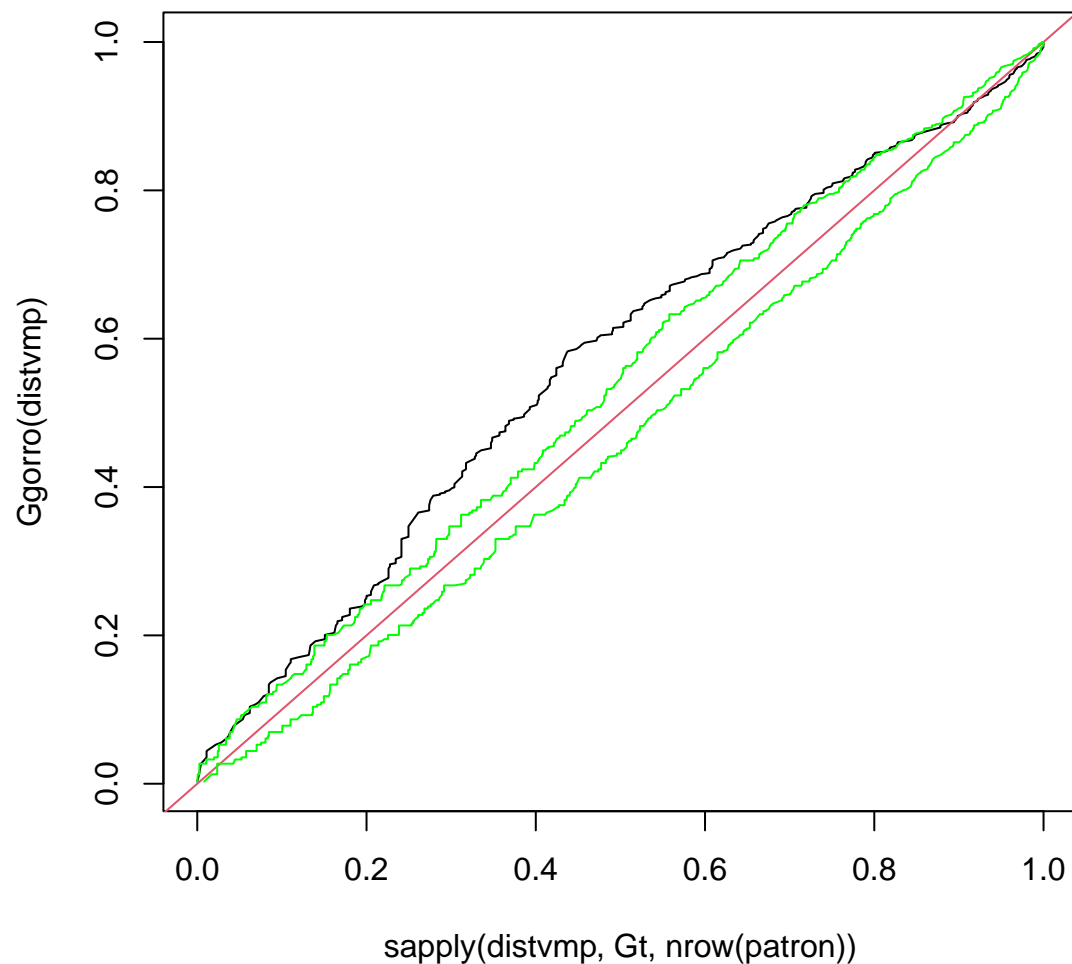
La comparación no arroja un resultado concluyente. Sin embargo, se observa que la línea empírica excede los límites superiores en distancias cortas, lo que implica que los puntos están más concentrados en áreas pequeñas en comparación con un patrón aleatorio. A mayores distancias, la línea cae por debajo de los límites esperados, sugiriendo áreas vacías más extensas que las previstas por un proceso aleatorio.

```
# La función calcdvmp() calcula las distancias al vecino más próximo
distvmp<-sort(calcdvmp(patron))
par(mfrow=c(1,2))
hist(distvmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Ggorro <- ecdf(distvmp)
plot(Ggorro, verticals= TRUE, do.p = F)
```

Distancias al vecino más próximo



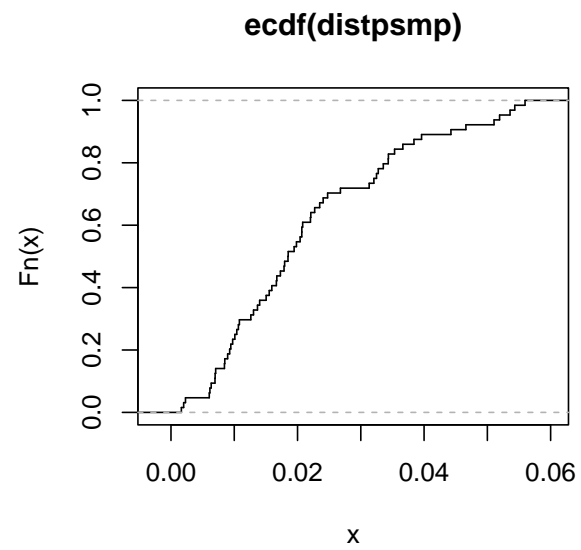
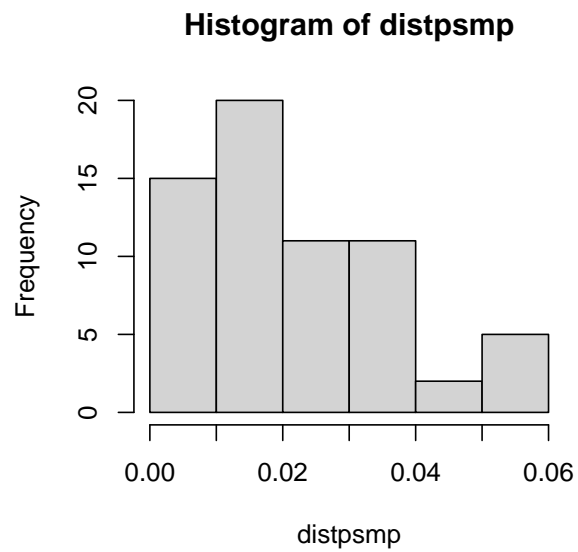
```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distvmp,Gt,nrow(patron)),Ggorro(distvmp),type="l")
abline(0,1,col=2)
lims<-Genvl(19,patron)
lines(sapply(lims$x1,Gt,nrow(patron)),Ggorro(distvmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Ggorro(distvmp),col="green")
```



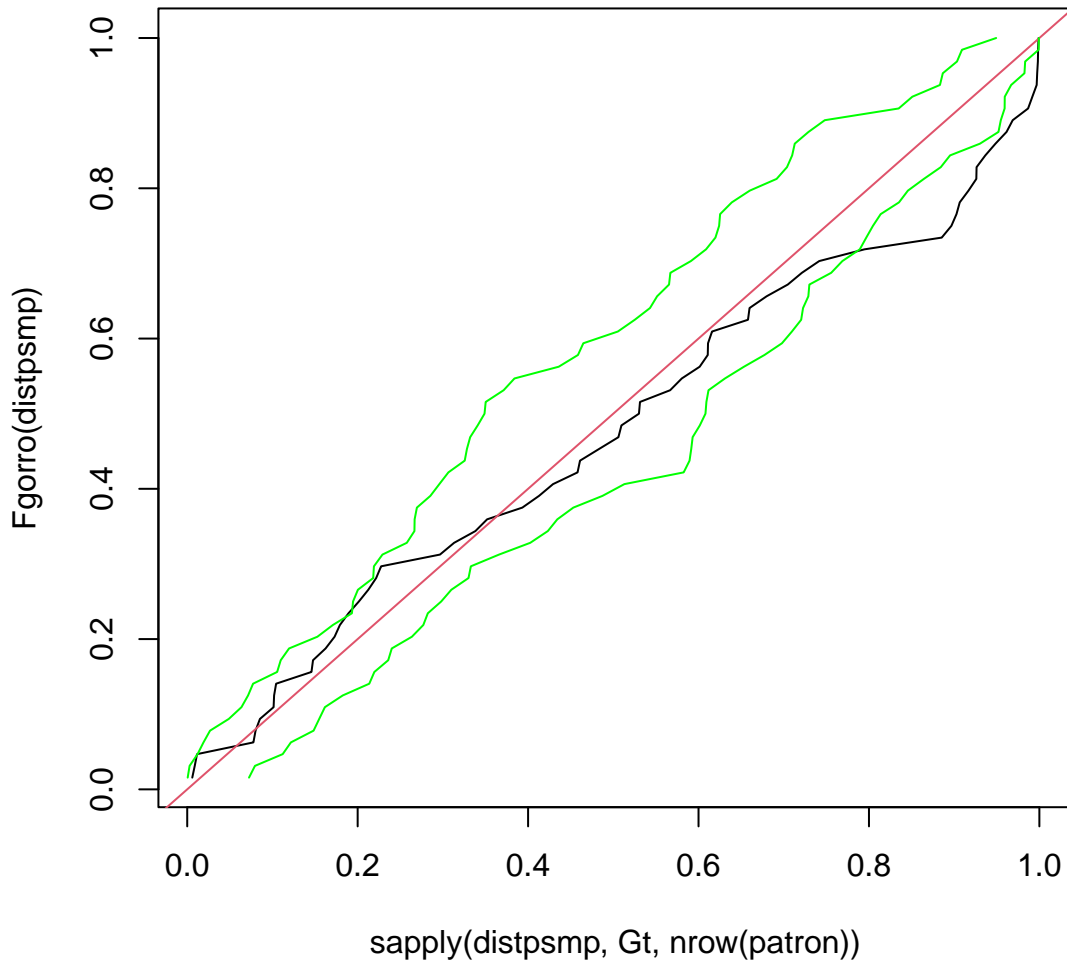
La función indica que el patrón crece mucho más rápidamente que un patrón aleatorio, con la línea empírica excediendo claramente los límites superiores. Esto refleja una alta frecuencia de distancias pequeñas entre puntos, lo que confirma la agrupación. Los puntos tienden a estar mucho más cerca unos de otros de lo que se esperaría bajo aleatoriedad completa.

```
# La función calcdpsmp() calcula las distancias desde 8*8 puntos aleatorios a los sucesos más próximos
distpsmp<-sort(calcdpsmp(patron,8))
par(mfrow=c(1,2))
hist(distpsmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Fgorro <- ecdf(distpsmp)
plot(Fgorro, verticals= TRUE, do.p = F)
```

Distancias de un punto al suceso más próximo



```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distpsmp,Gt,nrow(patron)),Fgorro(distpsmp),type="l")
abline(0,1,col=2)
lims<-Fenvl(19,patron,8)
lines(sapply(lims$xl,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
```



El análisis de la función revela que, aunque al principio la línea empírica está dentro de los límites simulados, termina excediéndolos por debajo en distancias mayores. Esto sugiere que hay áreas con menor densidad de puntos, posiblemente porque los puntos están concentrados en ciertas regiones, dejando otras más vacías. Este comportamiento es consistente con un patrón agrupado con áreas de alta densidad.

Conclusión

El análisis detallado realizado mediante diversos métodos confirma de manera concluyente que el patrón observado es agrupado y no aleatorio. Los resultados obtenidos a partir del análisis con grids y muestreo aleatorio de cuadrantes evidencian una notable variabilidad en la distribución de los puntos, respaldada por una varianza relativa elevada y por los valores de los índices de David-Moore, Lloyd y Morisita, los cuales señalan una fuerte tendencia al agrupamiento. Además, los métodos basados en distancias refuerzan esta interpretación: la función G revela que las distancias cortas son mucho más frecuentes de lo esperado bajo aleatoriedad, mientras que las funciones H y F reflejan la existencia de zonas densamente pobladas junto con áreas más dispersas, lo que caracteriza un patrón espacial marcadamente agrupado.

En resumen, los resultados descartan la hipótesis de aleatoriedad y confirman que los puntos se concentran en regiones específicas del espacio. El modelo más adecuado para describir este patrón es un proceso de

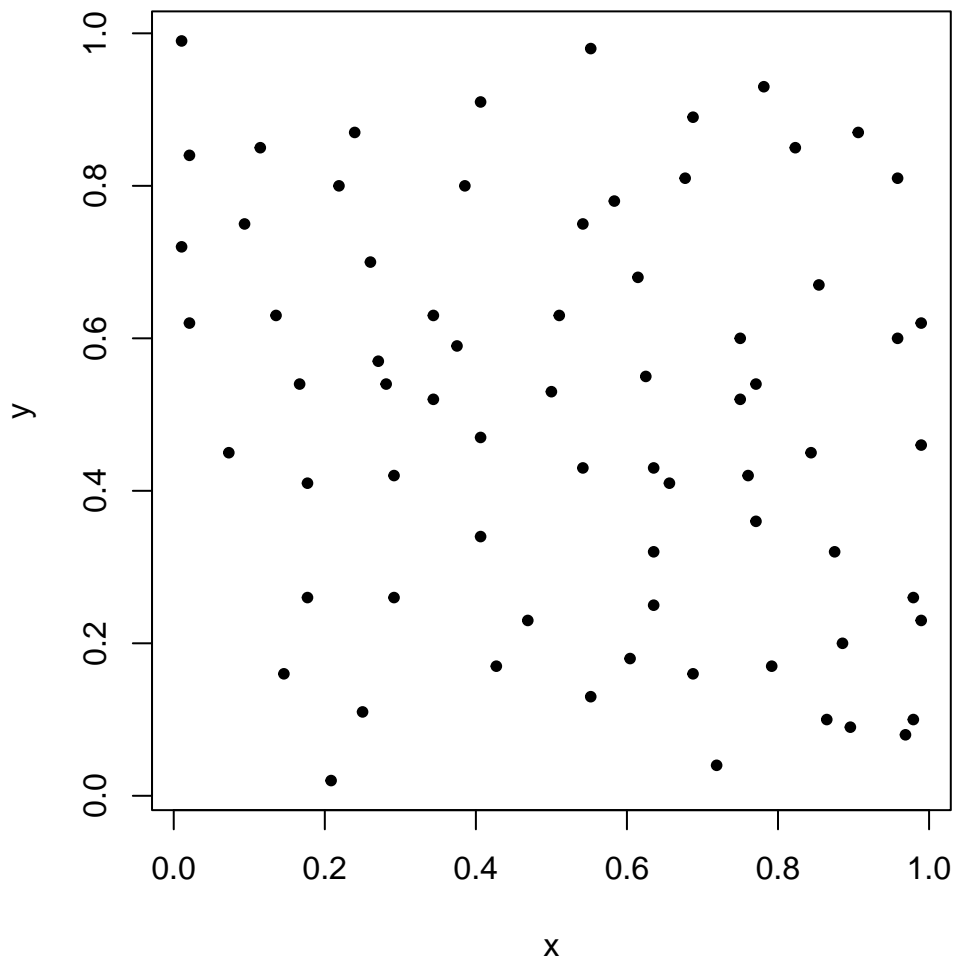
agrupación de Poisson si los agrupamientos están claramente definidos y no dependen de factores externos, o un proceso de Cox si estos agrupamientos están determinados por una intensidad espacial subyacente.

Patrón puntual Pino

```
patron <- pinos
```

Conteos por quadrats

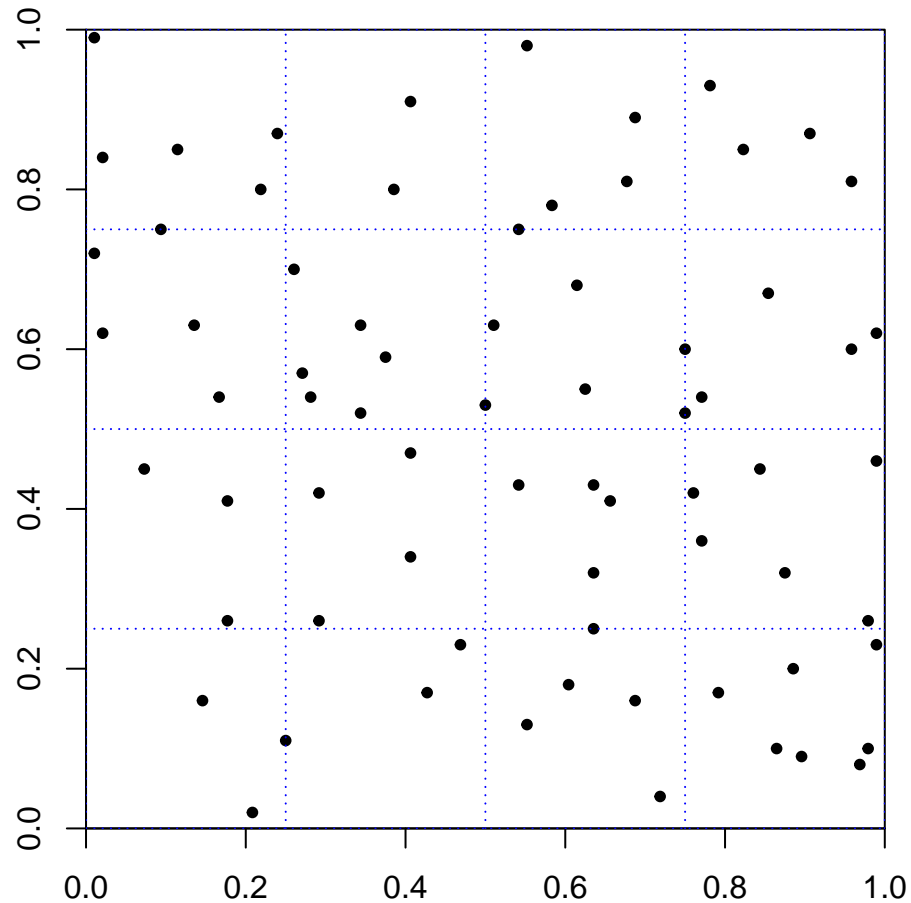
```
# Representación del patrón puntual en una región cuadrada  
par(pty='s')  
plot(patron,pch=20,xlab='x',ylab='y')
```



```
par(mfrow=c(1,1),pty="s")  
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)
```

```
# La función dibu.qua() dibuja las líneas de un grid de quadrats (4x4)
dibu.qua(4,4)
```

Análisis mediante un grid de quadrats.



```
# La función cuenta.qua() contabiliza cuantos sucesos hay en cada quadrat
nump.qua<-c(cuenta.qua(patron,4,4))
# El test chi-cuadrado de bondad de ajuste con los conteos observados y con todos los esperados iguales
chisq.test(nump.qua)
```

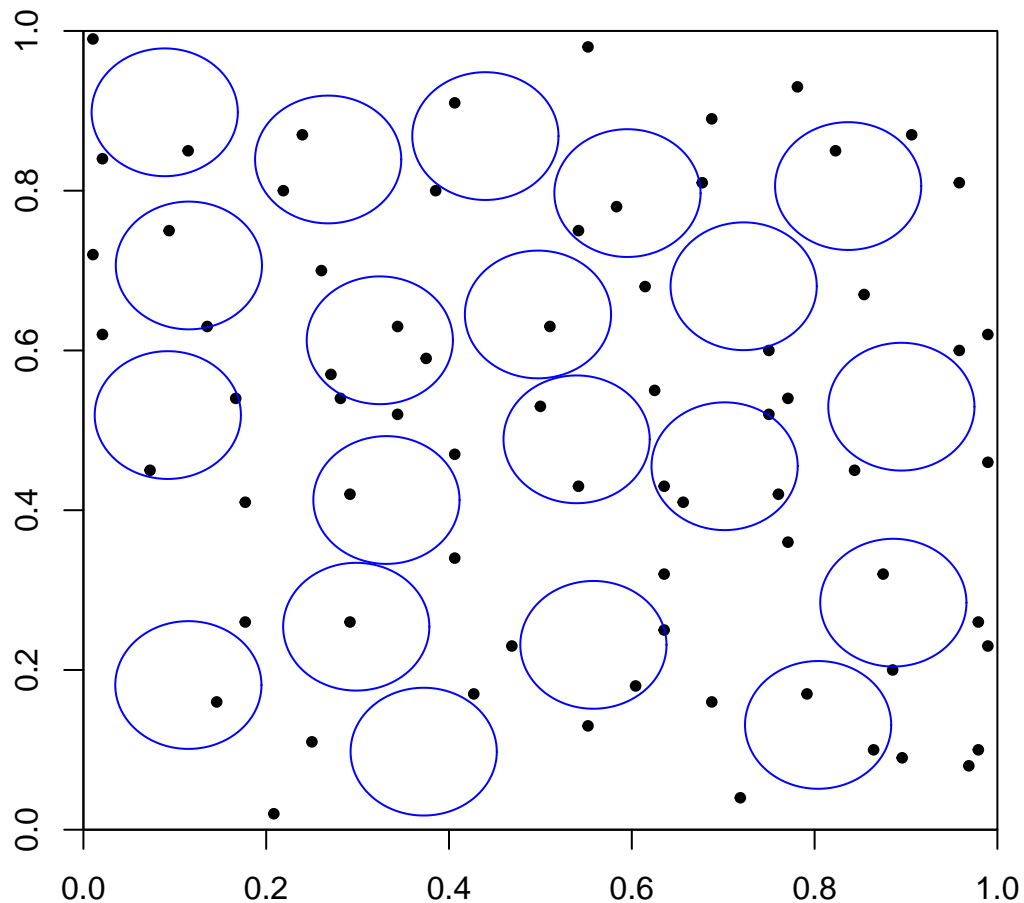
```
##
## Chi-squared test for given probabilities
##
## data:  nump.qua
## X-squared = 8.0986, df = 15, p-value = 0.9198
```

Dado que el p-valor es muy alto, no podemos rechazar la hipótesis de que este patrón sea completamente aleatorio. Esto sugiere que no hay diferencias significativas entre lo esperado y lo observado, y el patrón

parece ajustarse a una distribución aleatoria.

```
# La función Simqalea() simula los centros de 20 quadrats de radio 0.08
centros.qua<-Simqalea(20,0.08)
plot(patron,xlim=c(0,1),ylim=c(0,1),xlab="",ylab="",xaxs="i",yaxs="i",pch=20)
# La función dibalea.qua() dibuja los quadrats circulares simulados
dibalea.qua(centros.qua)
```

Análisis mediante un muestreo con quadrats aleatorios.



```
# La función cupenl.qua() calcula cuantos sucesos hay en cada quadrat
nump.qua<-cupenl.qua(patron,centros.qua)
# La función indices.qua() calcula los índices de agrupación
indices.qua(nump.qua)
```

```
## Índices de agrupamiento:
## -----
```

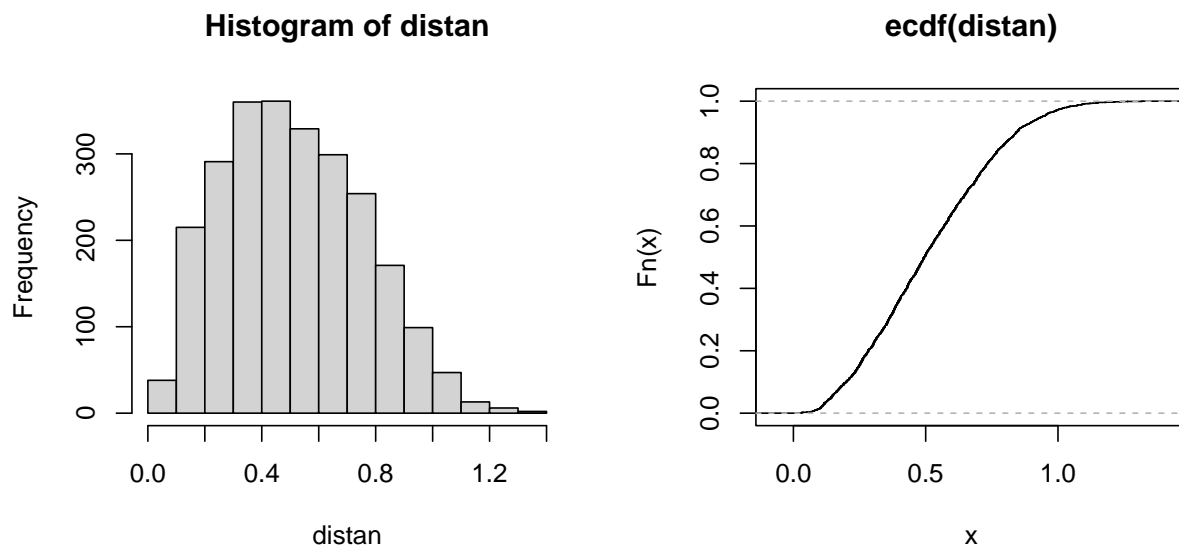
```
## Varianza Relativa: 0.556391
## Índice de David-Moore: -0.443609
## Desigualdad de Lloyd: 0.6831364
## Índice de Morisita: 0.03439153
```

El análisis de los puntos sobre los cuadrantes muestra una baja variabilidad, ya que la varianza relativa es 0.1700405, indicando que la varianza es considerablemente menor que la media. Esto sugiere un patrón cercano a la uniformidad. Los indicadores refuerzan esta interpretación: el índice de David-Moore (-0.8299595) muestra una tendencia hacia la regularidad, mientras que la desigualdad de Lloyd (0.3615696) confirma una distribución más homogénea que agrupada. El índice de Morisita (0.01846154), aunque sin referencia fija, sugiere un bajo nivel de agrupamiento en comparación con otros patrones.

Métodos de distancia

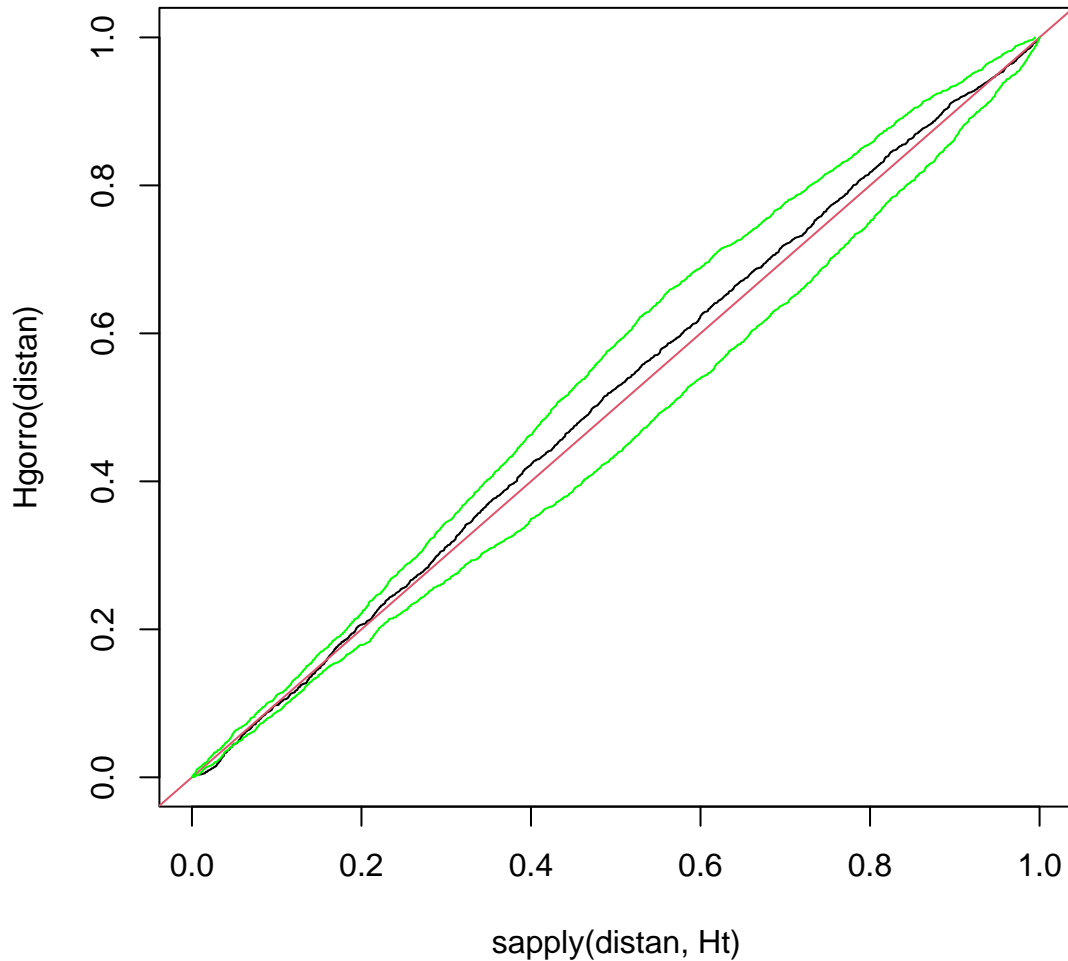
```
# Calculamos todas las distancias entre los sucesos
# La función genérica dist() crea una matriz (n,n) de distancias
# La función sort() ordena y convierte en vector las n*(n-1)/2 distancias
distan<-sort(dist(patron))
par(mfrow=c(1,2))
hist(distan)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Hgorro <- ecdf(distan)
plot(Hgorro, verticals= TRUE, do.p = F)
```

Distancias entre sucesos



```
# Compararemos la distribución empírica con la teórica Ht()
plot(sapply(distan,Ht),Hgorro(distan),type="l")
# La línea roja marca el ajuste perfecto con la aleatoriedad espacial completa
abline(0,1,col=2)
# La función Henvl() calcula 19 simulaciones de patrones completamente aleatorios
# y determina el mínimo y el máximo de la función H empírica para cada distancia
lims<-Henvl(19,patron)
# Se pintan en verde los límites inferior y superior de las simulaciones
```

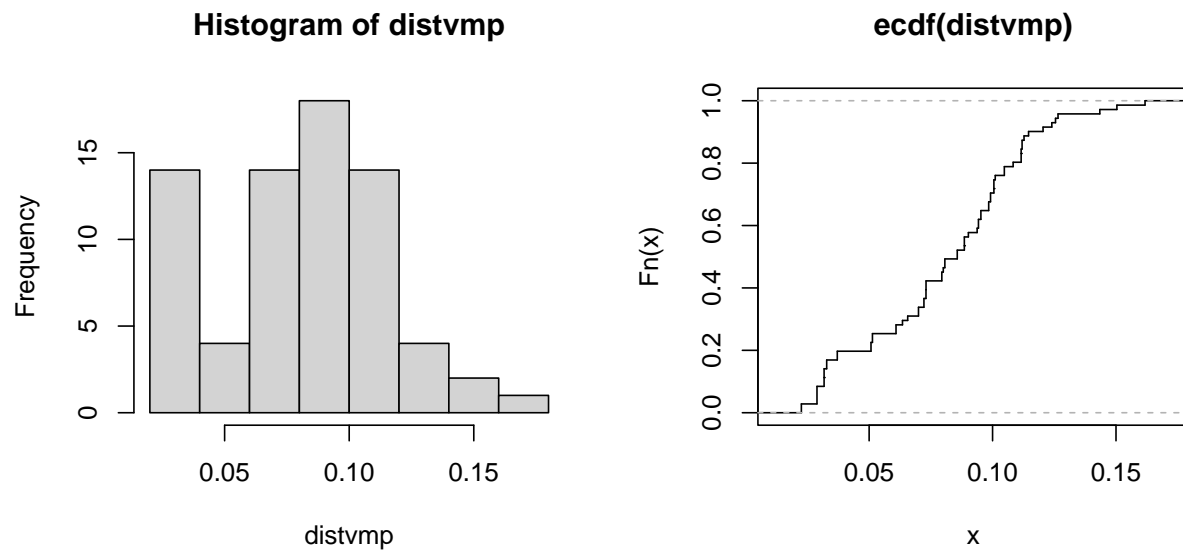
```
lines(sapply(lims$xl,Ht),Hgorro(distan),col="green")
lines(sapply(lims$xu,Ht),Hgorro(distan),col="green")
```



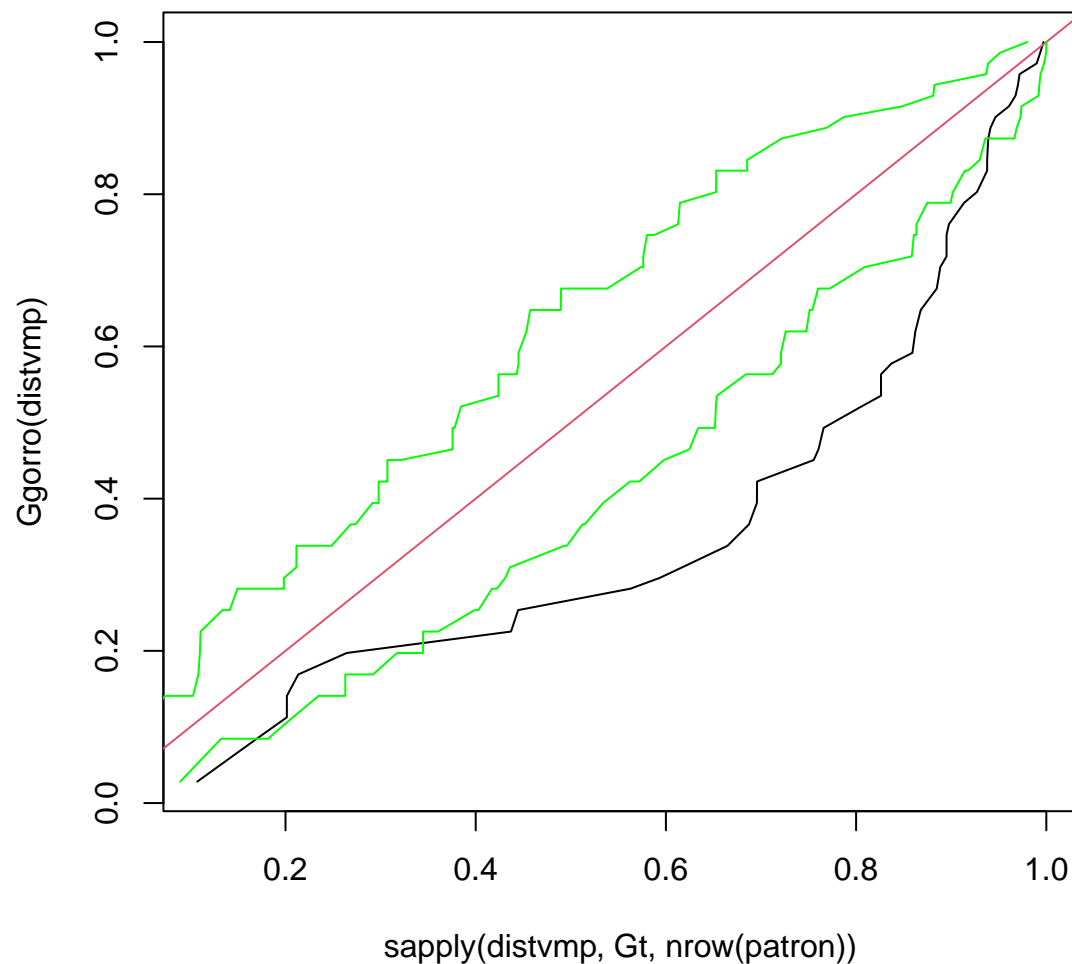
La función H muestra que la distribución empírica se ajusta casi perfectamente a las simulaciones teóricas, con la línea negra muy cercana a la roja y sin exceder los límites de las simulaciones en ningún momento significativo. Esto confirma la ausencia de patrones espaciales de agrupamiento o regularidad, apoyando la hipótesis de una distribución aleatoria.

```
# La función calcdvmp() calcula las distancias al vecino más próximo
distvmp<-sort(calcdvmp(patron))
par(mfrow=c(1,2))
hist(distvmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Ggorro <- ecdf(distvmp)
plot(Ggorro, verticals= TRUE, do.p = F)
```

Distancias al vecino más próximo



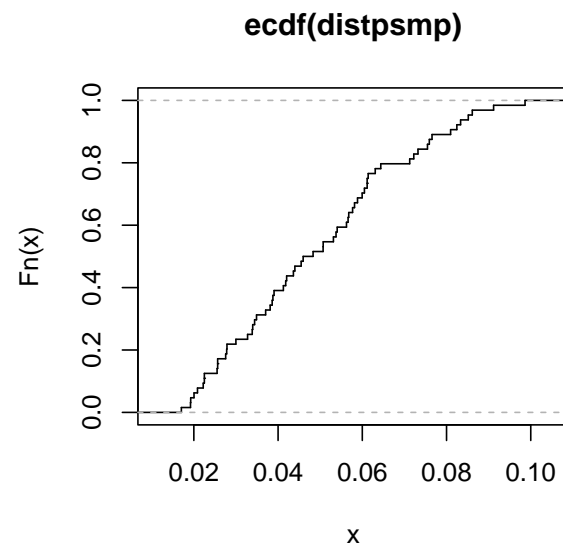
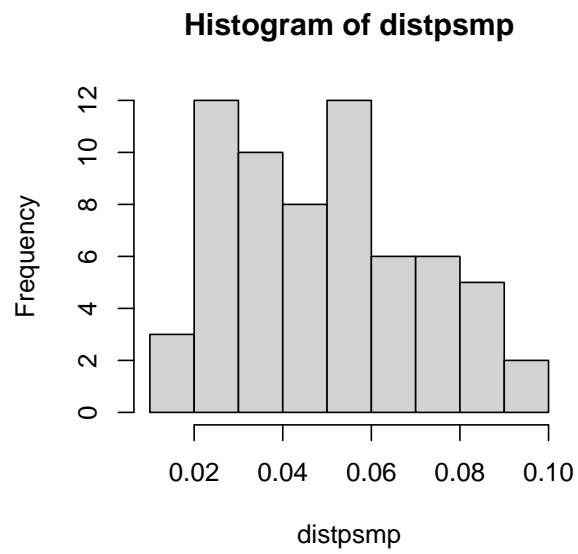
```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distvmp,Gt,nrow(patron)),Ggorro(distvmp),type="l")
abline(0,1,col=2)
lims<-Genvl(19,patron)
lines(sapply(lims$x1,Gt,nrow(patron)),Ggorro(distvmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Ggorro(distvmp),col="green")
```



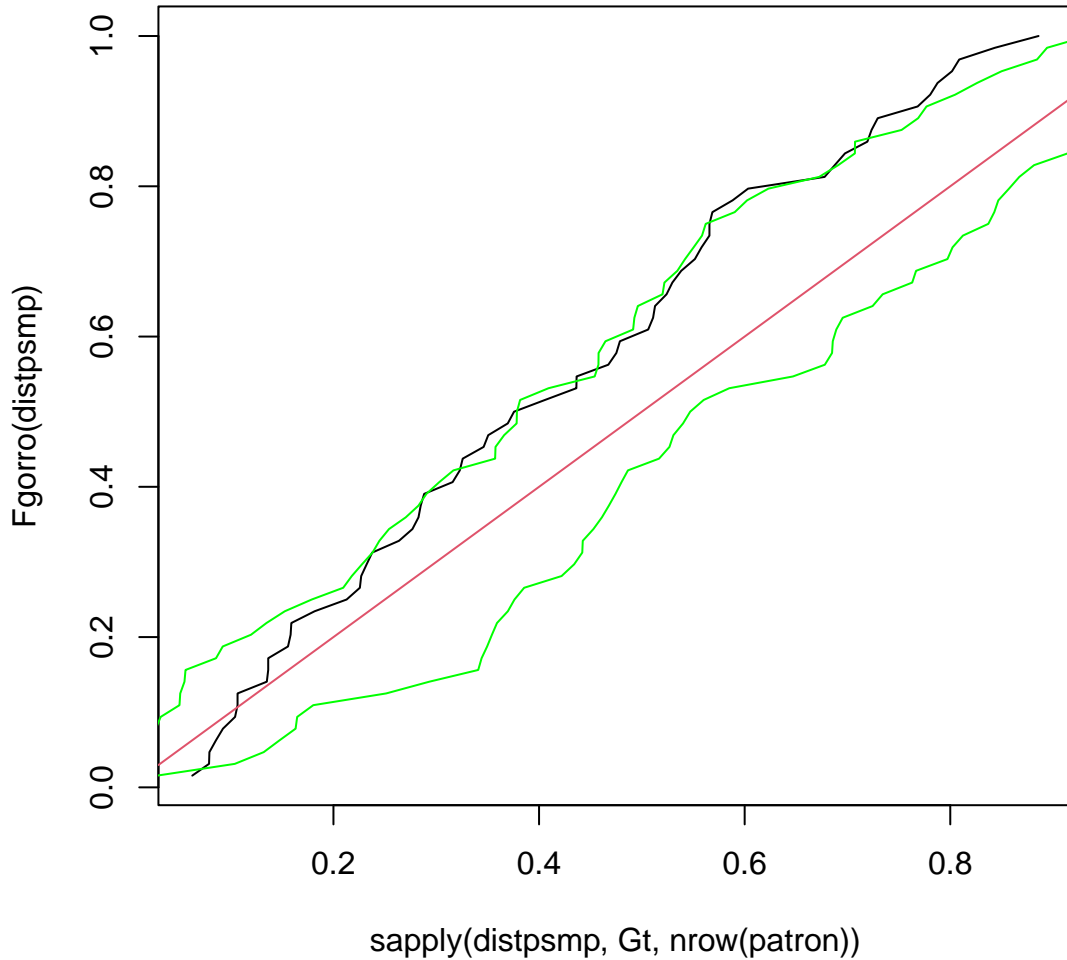
En el caso de la función G, la línea empírica cae por debajo de las simulaciones aleatorias, lo que sugiere una leve tendencia hacia una mayor dispersión de los puntos en distancias cortas. Esto podría ser indicativo de un comportamiento cercano a la regularidad, aunque el efecto no es suficientemente fuerte como para rechazar la aleatoriedad completa.

```
# La función calcdpsmp() calcula las distancias desde 8*8 puntos aleatorios a los sucesos más próximos
distpsmp<-sort(calcdpsmp(patron,8))
par(mfrow=c(1,2))
hist(distpsmp)
# La función genérica ecdf() calcula la función de distribución acumulada empírica
Fgorro <- ecdf(distpsmp)
plot(Fgorro, verticals= TRUE, do.p = F)
```

Distancias de un punto al suceso más próximo



```
# Compararemos la distribución empírica con la teórica Gt()
plot(sapply(distpsmp,Gt,nrow(patron)),Fgorro(distpsmp),type="l")
abline(0,1,col=2)
lims<-Fenvl(19,patron,8)
lines(sapply(lims$xl,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
lines(sapply(lims$xu,Gt,nrow(patron)),Fgorro(distpsmp),col="green")
```



La función F muestra un comportamiento inicial dentro de los límites de las simulaciones aleatorias, pero al final excede ligeramente los límites inferiores. Esto podría reflejar una menor densidad de puntos en ciertas áreas del espacio, aunque no es suficientemente pronunciado para desviar la conclusión de aleatoriedad.

Conclusión

El análisis detallado a través de diferentes métodos confirma que el patrón observado es principalmente aleatorio, aunque con ligeros indicios de regularidad en ciertas métricas. Los métodos basados en grids y muestreo con quadrats muestran que la distribución no presenta agrupamientos significativos ni una variabilidad destacada entre cuadrantes, lo que indica que los puntos están distribuidos de manera independiente y uniforme. La baja varianza relativa, junto con el índice de David-Moore negativo y los valores bajos de los índices de Lloyd y Morisita, refuerzan esta idea al señalar una tendencia hacia la homogeneidad, más que hacia la agrupación o un patrón totalmente regular.

Por otro lado, los análisis basados en distancias corroboran esta conclusión. La función H muestra una alta concordancia entre la distribución empírica y las simulaciones aleatorias, lo que implica que no hay una estructura espacial evidente. La función G , aunque refleja una ligera tendencia hacia la dispersión en distancias cortas (línea empírica por debajo de las simulaciones), no lo hace de manera suficientemente

marcada como para considerar el patrón como regular. Finalmente, la función F sugiere que en ciertas áreas los puntos podrían estar ligeramente más dispersos de lo esperado bajo aleatoriedad, aunque este comportamiento no es dominante en el patrón general.

En conjunto, los resultados destacan que el patrón no muestra indicios claros de agrupamiento ni de regularidad significativa, lo que respalda la hipótesis de una distribución aleatoria.

El modelo más adecuado para representar este tipo de patrón sería un proceso de Poisson homogéneo, ya que describe con precisión una distribución de puntos que ocurren de manera independiente entre sí y con la misma probabilidad en todo el espacio. Aunque existen ligeras desviaciones hacia la regularidad en distancias cortas, estas no son lo suficientemente fuertes como para justificar un modelo más complejo.