Estatística Espacial

Grupo

- Americo Freitas
- André Martins
- Antonio Junior

Professor

- Jony Arrais Pinto Junior

Descrição

- Trabalho 2

Dados – Localizações das ocorrências de crimes na cidade de Houston.

Análise – Faça uma análise descritiva/exploratória (apresentação do espaço, do fenômeno de interesse, além de investigar os efeitos de 1a e 2a ordem) das localizações das ocorrências dos crimes de acordo com seus tipos (variável offense). Ao realizar a análise discutam a estimação da função de intensidade e diferentes formas de verificar a existência do efeito de segunda ordem. Discutam as possíveis diferenças observadas entre os efeitos estimados (1a e 2a ordem) dos 7 cenários avaliados.

Carregando as bibliotecas necessárias para a análise do modelo

Lendo o shapefile com os bairros de Houston no formato Lat e Long

```
houston_shp = readOGR("Houston_City_Limit/Houston_City_Limit.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\temp\Trab_Estatistica_Espacial\Houston_City_Limit\Houston_City_Limit.shp", lay
er: "Houston_City_Limit"
## with 743 features
## It has 22 fields
## Integer64 fields read as strings:  OBJECTID
```

```
houston=houston_shp
#houston=spTransform(houston_shp, CRS("+proj=longlat +datum=WGS84"))
```

Carregando a Base Houston (Ocorrencias), e as bases auxiliares separadas por delito.

Bases Auxilares:

- Robbery, Aggravated Assault, Auto Theft, Burglary, Rape, Murder, Hours

```
#class(houston)

houston_limit<-read.csv2("Base Houston.csv", sep = ",", dec = "." )
houston_limit
```

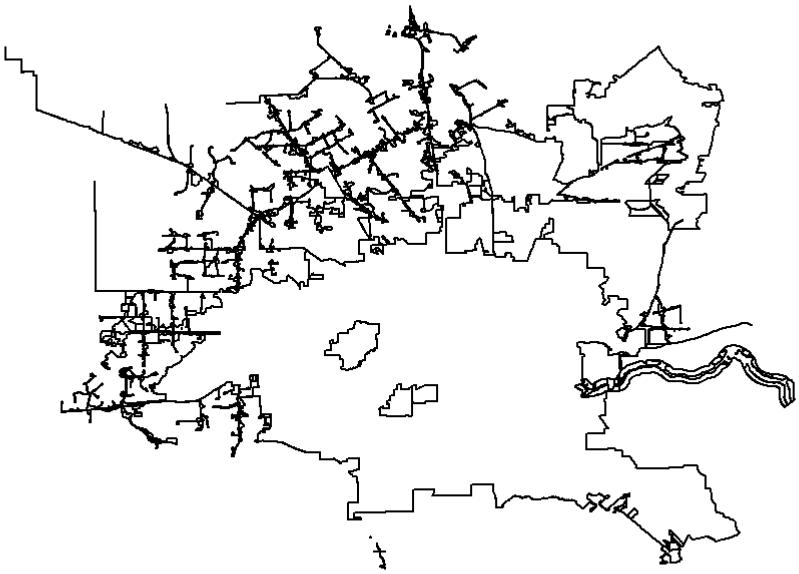
```
##                time      date hour premise            offense     beat
## 1    01/01/10 06:00  01/01/10    0     18A             murder 1,50E+31
## 2    05/01/10 21:00  01/05/10   15     20A             murder     6B60
## 3    06/01/10 04:00  01/05/10   22     18A             murder    19G50
## 4    07/01/10 00:00  01/06/10   18     20R             murder     6B40
## 5    07/01/10 00:00  01/06/10   18     20R             murder     8C50
## 6    07/01/10 07:00  01/07/10    1     13R             murder    10H50
## 7    07/01/10 21:00  01/07/10   15     20A             murder     8C20
## 8    09/01/10 04:00  01/08/10   22     20R             murder    11H30
## 9    09/01/10 20:00  01/09/10   14     13R             murder     5F20
## 10   11/01/10 05:00  01/10/10   23     20R             murder     2A50
## 11   16/01/10 04:00 1/15/2010   22     20R             murder    10H60
## 12   18/01/10 00:00 1/17/2010   18     18C             murder    10H50
## 13   18/01/10 20:00 1/18/2010   14     13R             murder    10H40
## 14   18/01/10 22:00 1/18/2010   16      70             murder     8C20
## 15   22/01/10 21:00 1/22/2010   15     13R             murder    19G10
## 16   02/02/10 10:00  02/02/10    4     20A             murder    13D40
## 17   03/02/10 02:00  02/02/10   20     20A             murder     1A50
## 18   04/02/10 00:00  02/03/10   18     20A             murder    20G10
```

```
#------------variavies por delito-----------------------------------

hrobbery<-read.csv2("robbery.csv",sep = ";", dec = ".")
hagg<- read.csv2("aggravated assault.csv", dec = ".")
hauto<- read.csv2("auto theft.csv", dec = ".")
hburglary<- read.csv2("burglary.csv", dec = ".")
hrape<- read.csv2("rape.csv", dec = ".")
hmurder<-read.csv2("murder.csv", dec = ".")
write.csv2(houston_limit, "hous.csv")

#ls()
#------------------------------------------------------------------
```

Análise do mapa

```
#Definindo o shapefile como uma janela onde os pontos serao plotados
HCL <- as.owin(houston)
#Plotando o shapefile
plot(HCL)
```

HCL



```
#Criando o padrao de pontos a ser plotado
Houston_ppp = ppp(houston_limit$lon, houston_limit$lat,window=HCL )
```

```
## Warning: 45 points were rejected as lying outside the specified window
```

```
## Warning: data contain duplicated points
```

```
#---------------criando pontos por delitos------------------------------
hagg_ppp = ppp(hagg$lon, hagg$lat,window=HCL )
```

```
## Warning: 8 points were rejected as lying outside the specified window

## Warning: data contain duplicated points
```

```
hrobbery_ppp = ppp(hrobbery$lon, hrobbery$lat,window=HCL )
```

```
## Warning: 10 points were rejected as lying outside the specified window

## Warning: data contain duplicated points
```

```
hmurder_ppp = ppp(hmurder$lon,hmurder$lat,window=HCL )
```

```
## Warning: 2 points were rejected as lying outside the specified window

## Warning: data contain duplicated points
```

```
hauto_ppp = ppp(hauto$lon, hauto$lat,window=HCL )
```

```
## Warning: 12 points were rejected as lying outside the specified window

## Warning: data contain duplicated points
```

```
hrape_ppp = ppp(hrape$lon, hrape$lat,window=HCL )
```

```
## Warning: 7 points were rejected as lying outside the specified window

## Warning: data contain duplicated points
```

```
hburglary_ppp = ppp(hburglary$lon, hburglary$lat,window=HCL )
```

```
## Warning: 6 points were rejected as lying outside the specified window

## Warning: data contain duplicated points
```

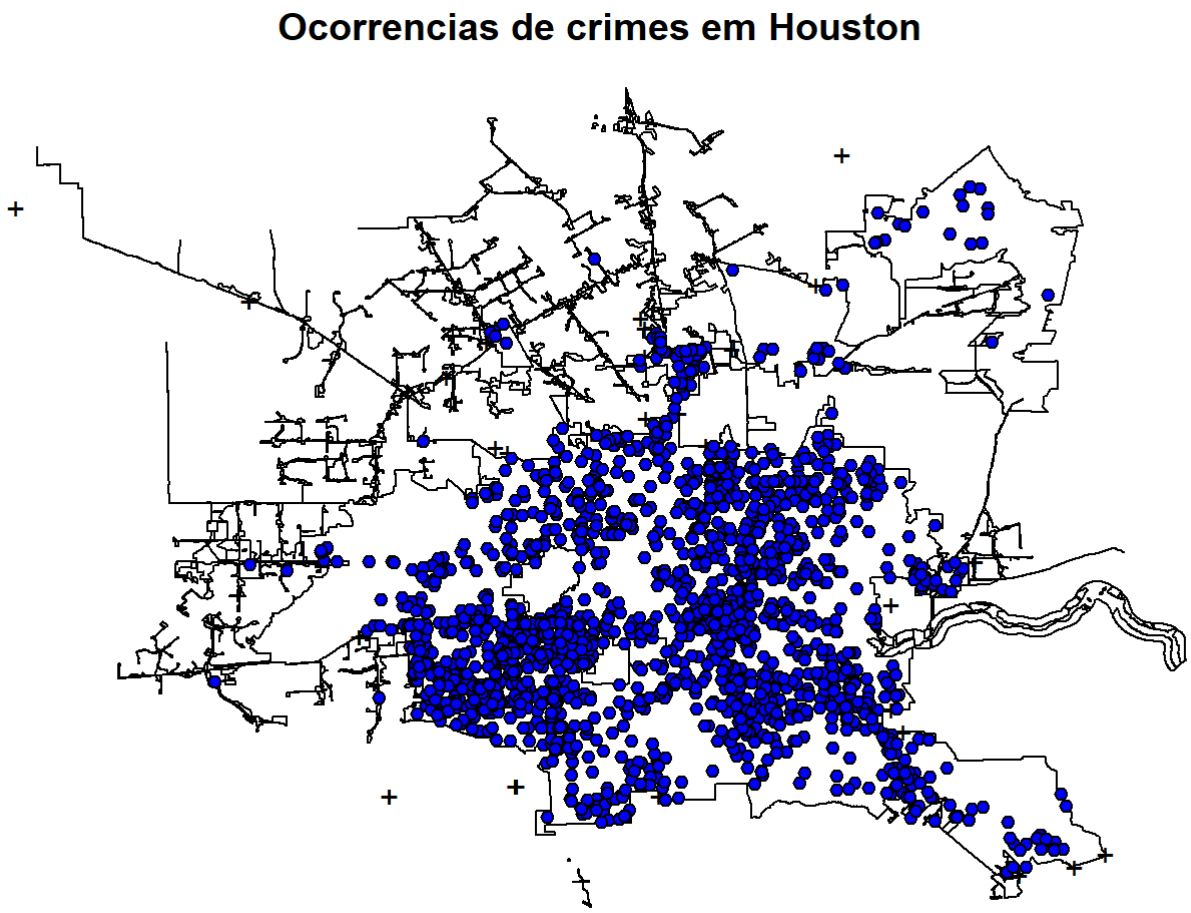
```
#---------------------------------------------------------------------
```

Ocorrências de crimes em Houston

Ao observar o mapa, concluímos que existem áreas com maior concentração de ocorrências, sugerindo que os eventos ocorram mais ao centro.

```
##ppp - criar um objeto com classe ppp representando o padrao de pontos
#Argumentos:
#x - longitude
#y - latitude
#window - um objeto owin.

#Plotando as localizacoes dos delitos
par(mar=c(0.5,0.5,1.5,0.5))

plot(Houston_ppp, pch=21, cex=0.9, bg="blue", main="Ocorrencias de crimes em Houston")
```

```
## Warning in plot.ppp(Houston_ppp, pch = 21, cex = 0.9, bg = "blue", main =
## "Ocorrencias de crimes em Houston"): 45 illegal points also plotted
```

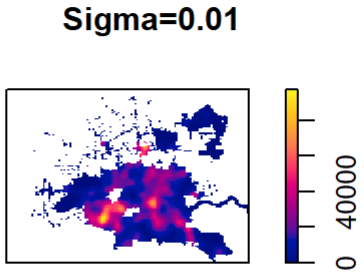


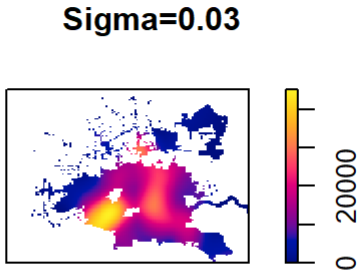
Efeitos de primeira Ordem

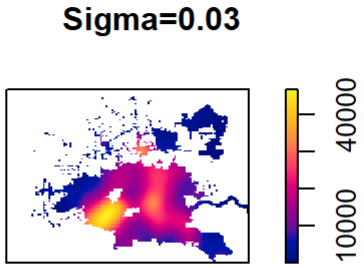
Análise geral

Ao aplicar o raio dos estimadores com pesos distintos reparamos que para o maior raio (0,03) utilizando o kernel Quártico, temos uma maior suavidade visual nas áreas com menos ocorrências próximas as áreas de maior ocorrência, o que intensifica o alerta visual fornecido pelo mapa. Para o raio menor (0,01) no kernel Normal (gaussiano), conseguimos identificar bem as areas com buracos de ocorrência dos eventos. Para alertar uma situação de perigo, eu avalio o de maior raio (0,03) Quártico como mais adequado.

```
#-----------------PRIMEIRA ORDEM-----------------------------------------------------------
-------
#PLOTANDO GRAFICO EM  EFEITOS EM PRIMEIRA ORDEM
Houston.g = density.ppp(x = Houston_ppp, sigma=0.01, kernel="gaussian")
Houston.q = density.ppp(x = Houston_ppp, sigma=0.01, kernel="quartic")
#Plotando os dados e as funcoes intensidades estimadas pelas diversas funcoes kernel
par(mfrow=c(2,2))

plot(density.ppp(Houston_ppp, sigma=0.01, kernel="gaussian"), main="Sigma=0.01", cex.main=0.5
)#gaussian
plot(density.ppp(Houston_ppp, sigma=0.03, kernel="quartic"), main="Sigma=0.03", cex.main=0.5)
#quartic
plot(density.ppp(Houston_ppp, sigma=0.03, kernel="gaussian"), main="Sigma=0.03", cex.main=0.5
)#quartic
##density.ppp - calcula a funcao de intensidade de acordo com o kernel escolhido
#Argumentos:
#x - objeto da classe ppp
#sigma - é o valor do raio (tau na expressao dos slides)
#kernel - o kernel que deseja-se usar
```



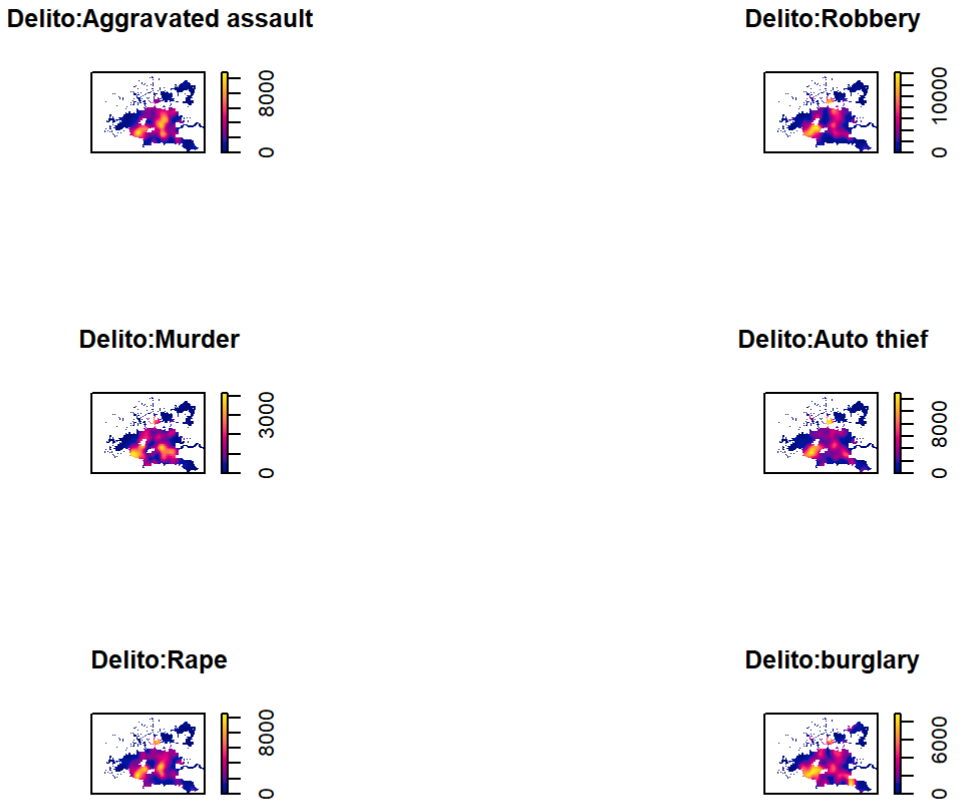




Análise por delito

Ao analisar cada delito separadamente, identificamos que as regiões de predominancia de ocorrências são razoavelmente comuns idependente do delito.

```
#----------------PLOTANDO GRAFICO EM  EFEITOS EM PRIMEIRA ORDEM POR DELITO

par(mfrow=c(3,2))
plot(density.ppp(hagg_ppp , sigma=0.02, kernel="gaussian"), main="Delito:Aggravated assault",
 cex.main=0.5)#quartic
plot(density.ppp(hrobbery_ppp , sigma=0.02, kernel="gaussian"), main="Delito:Robbery", cex.ma
in=0.5)#quartic
plot(density.ppp(hmurder_ppp , sigma=0.02, kernel="gaussian"), main="Delito:Murder", cex.main
=0.5)#quartic
plot(density.ppp(hauto_ppp , sigma=0.02, kernel="gaussian"), main="Delito:Auto thief", cex.ma
in=0.5)#quartic
plot(density.ppp(hrape_ppp  , sigma=0.02, kernel="gaussian"), main="Delito:Rape", cex.main=0.
5)#quartic
plot(density.ppp(hburglary_ppp , sigma=0.02, kernel="gaussian"), main="Delito:burglary", cex.
main=0.5)#quartic
```




```
par(mar=c(0.5,0.5,1.5,0.5))
```

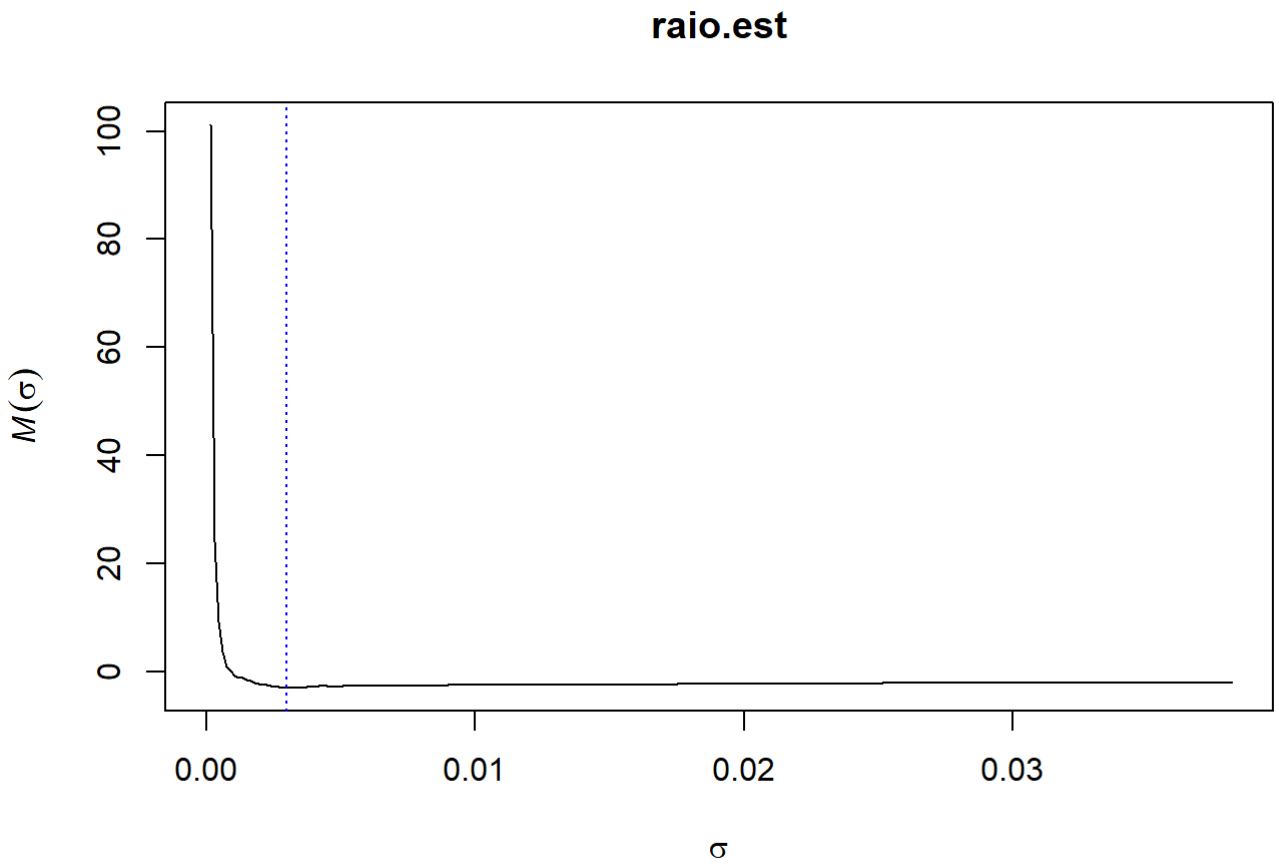
Análise por crimes

Ao analisar os pontos por validação cruzada, para suavisação do kernel, a largura de banda estimada é de aproximadamente 0.0029.

```
#--------------------plotando graficos por crimes------------------------------

#Plotando o grafico com recursos do Google Maps
#Criando o grafico com a densidade e o layout do Google Maps
#HT <- get_map("Houston, Texas", zoom=8,maptype='hybrid')
#plot(HT)
#google = plot(HT) + stat_density2d(aes(x=lat,y=lon, fill = ..level..),alpha = .8, h=.025, n
 = 400,geom = "polygon", data = houston_limit)
#plot(google)
#google+ scale_fill_gradient(low = "black", high= "red") + facet_wrap(~ offense)

#Funcao que estima o raio por meio de validacao cruzada (custosa computacionalmente)
raio.est = bw.diggle(Houston_ppp)
raio.est
```

```
##       sigma
## 0.002992444
```

```
plot(raio.est)
```



Efeitos de segunda Ordem

Estudando o efeito de segunda ordem através de uma amostra de 50 ocorrências de cada delito, selecionadas aleatóriamente para reduzir o custo computacional.

```
#-----------------------------------SEGUNDA ORDEM---------------------------------
#Sorteando uma amostra de tamanho 100 para estudar o efeito de segunda ordem por conta do cus
to computacional
len=50
aHouston = sample_n(houston_limit,len)
ahagg = sample_n(hagg ,len)
ahrobbery = sample_n(hrobbery,len)
ahmurder = sample_n(hmurder,len)
ahauto= sample_n(hauto,len)
ahrape= sample_n(hrape,len)
ahburglary= sample_n(hburglary,len)

aHouston_ppp<-ppp(aHouston$lon, aHouston$lat, window=HCL)
ahagg_ppp<-ppp(ahagg$lon, ahagg$lat, window=HCL)
```

```
## Warning: 1 point was rejected as lying outside the specified window
```

```
ahrobbery_ppp<-ppp(ahrobbery$lon,ahrobbery$lat, window=HCL)
```

```
## Warning: 1 point was rejected as lying outside the specified window
```

```
## Warning: data contain duplicated points
```

```
ahmurder_ppp<-ppp(ahmurder$lon, ahmurder$lat, window=HCL)
```

```
## Warning: 1 point was rejected as lying outside the specified window

## Warning: data contain duplicated points
```

```
ahauto_ppp <-ppp(ahauto$lon, ahauto$lat, window=HCL)
```

```
## Warning: 1 point was rejected as lying outside the specified window
```

```
ahrape_ppp <-ppp(ahrape$lon, ahrape$lat, window=HCL)
ahburglary_ppp <-ppp(ahburglary$lon,ahburglary$lat, window=HCL)

par(mfrow = c(4,2))
par(mar=c(0.5,0.5,1.5,0.5))
plot(ahagg_ppp, pch=21, cex=0.9, bg="blue", main="Aggravated assault")
```

```
## Warning in plot.ppp(ahagg_ppp, pch = 21, cex = 0.9, bg = "blue", main =
## "Aggravated assault"): 1 illegal points also plotted
```

```
plot(ahrobbery_ppp, pch=21, cex=0.9, bg="blue", main="Robbery")
```

```
## Warning in plot.ppp(ahrobbery_ppp, pch = 21, cex = 0.9, bg = "blue", main =
## "Robbery"): 1 illegal points also plotted
```

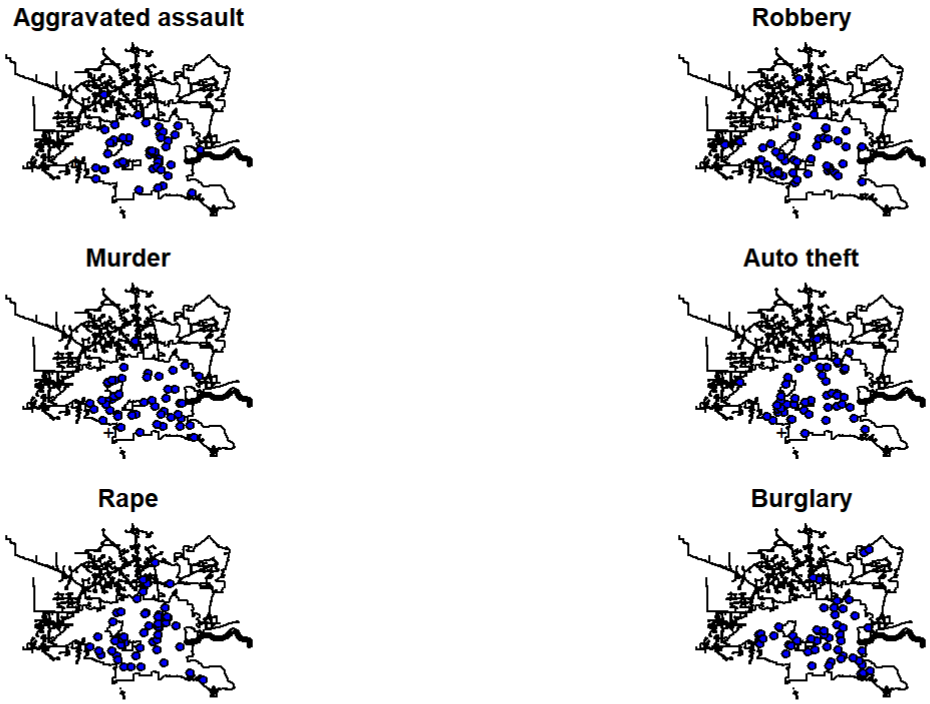
```
plot(ahmurder_ppp, pch=21, cex=0.9, bg="blue", main="Murder")
```

```
## Warning in plot.ppp(ahmurder_ppp, pch = 21, cex = 0.9, bg = "blue", main =
## "Murder"): 1 illegal points also plotted
```

```
plot(ahauto_ppp, pch=21, cex=0.9, bg="blue", main="Auto theft")
```

```
## Warning in plot.ppp(ahauto_ppp, pch = 21, cex = 0.9, bg = "blue", main =
## "Auto theft"): 1 illegal points also plotted
```

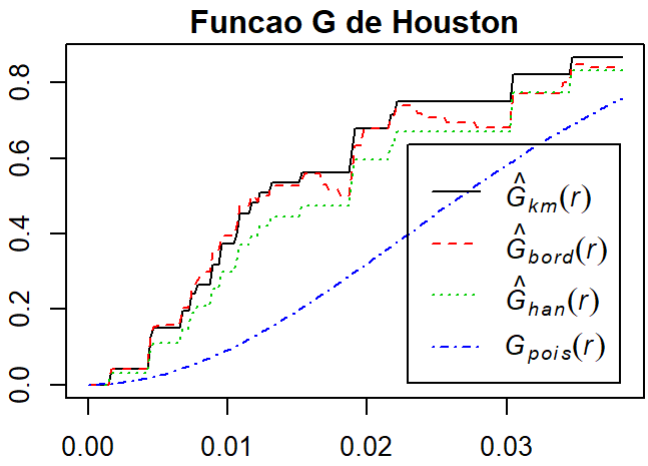
```
plot(ahrape_ppp, pch=21, cex=0.9, bg="blue", main="Rape")
plot(ahburglary_ppp, pch=21, cex=0.9, bg="blue", main="Burglary")
```

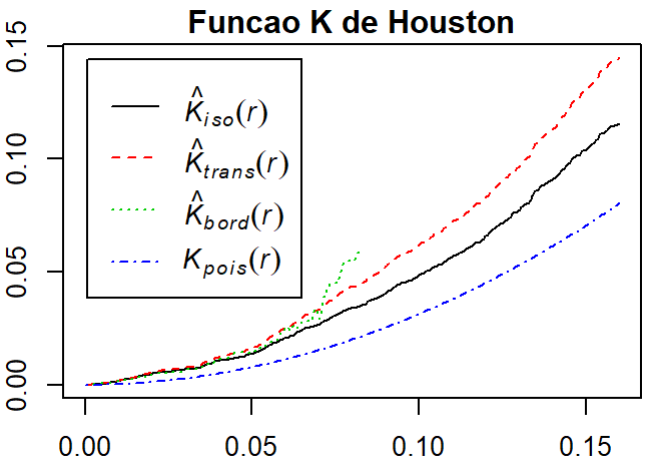


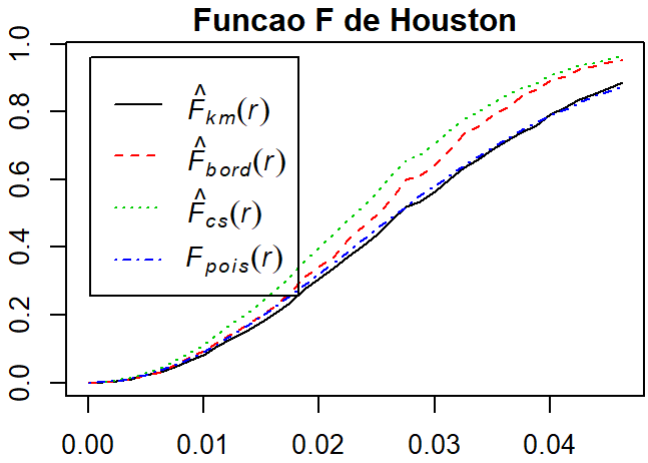
Observando interação entre os eventos

Ao observar a interação entre os eventos, aplicamos para todos os tipos de delitos as funções G, K e F. Em todos os casos, observamos que no resultado das funções G e K, as linhas estão a cima da curva teórica. E para o resultado da função F as linhas estão abaixo da curva teórica. Em todas as tres funções, confirmamos nossa percepção de que a interação entre os eventos é de agrupamento.

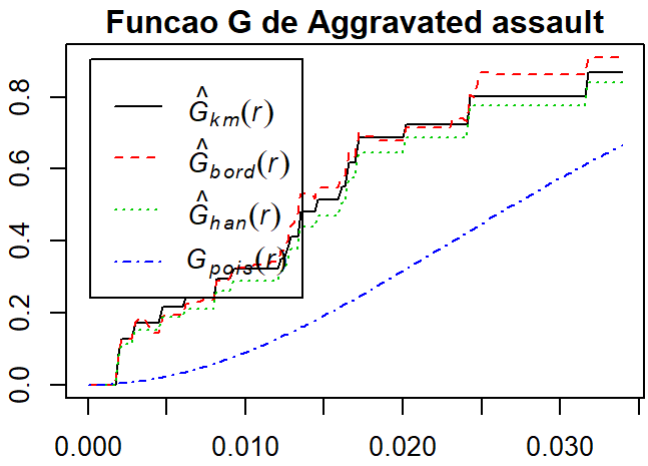
```
#-----------Plotando as funcoes G, K e F de Houston-------------------------
par(mfrow = c(2,2))
par(mar=c(2.5,2.5,1.5,.5))

HC.G = Gest(aHouston_ppp)
HC.K = Kest(aHouston_ppp)
HC.F = Fest(aHouston_ppp)

plot(HC.G, main="Funcao G de Houston")
plot(HC.K, main="Funcao K de Houston")
plot(HC.F, main="Funcao F de Houston")
```

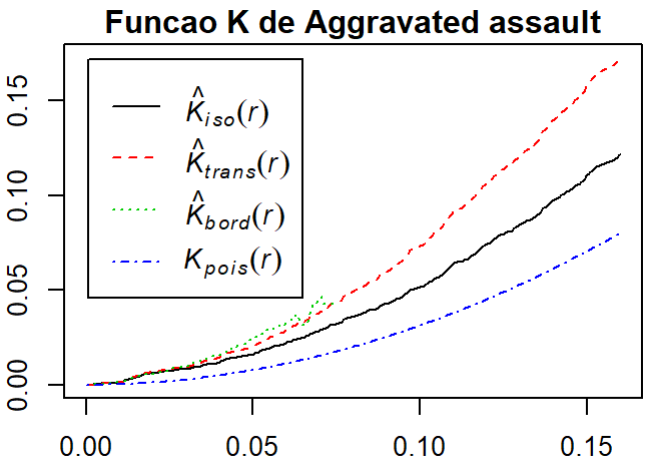


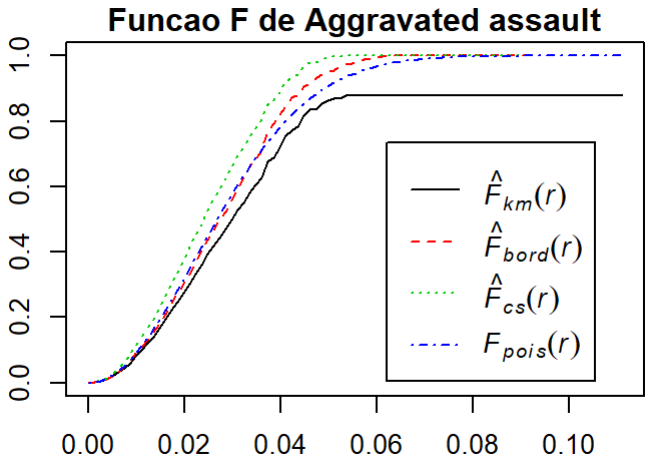




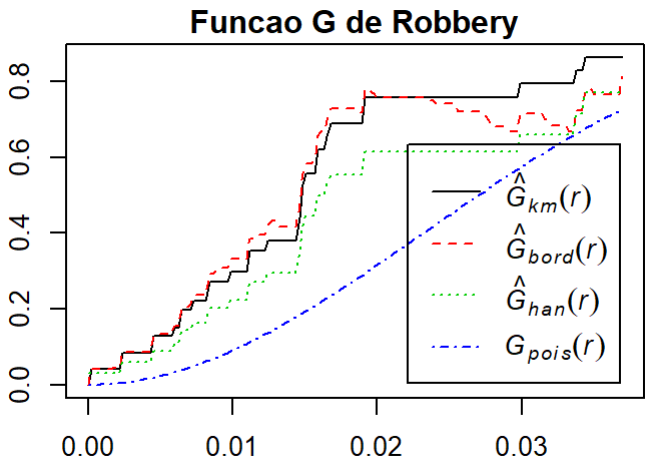

```
#-------------Plotando as funcoes G, K e F dos delitos----------------------

KHagg.G=Gest(ahagg_ppp)
KHagg.K=Kest(ahagg_ppp)
KHagg.F=Fest(ahagg_ppp)

par(mfrow = c(2,2))
par(mar=c(2.5,2.5,1.5,.5))
plot(KHagg.G, main="Funcao G de Aggravated assault")
plot(KHagg.K, main="Funcao K de Aggravated assault")
plot(KHagg.F, main="Funcao F de Aggravated assault")
```

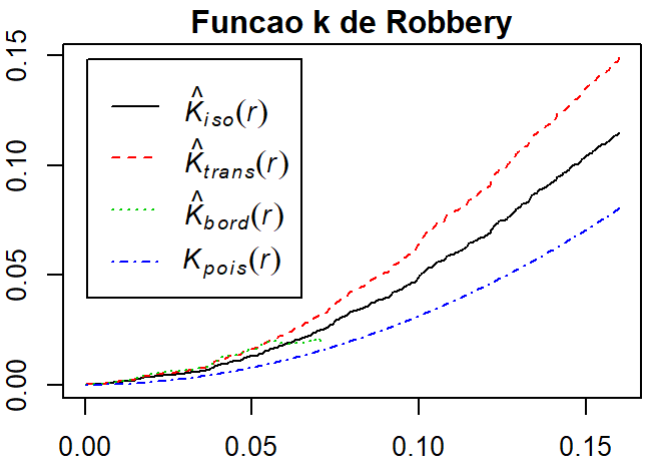


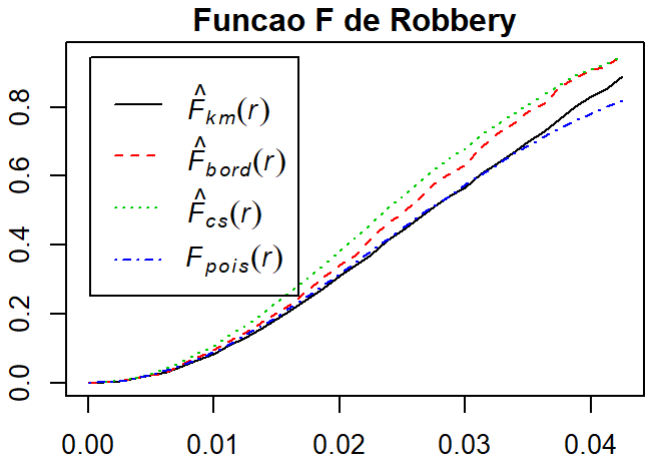




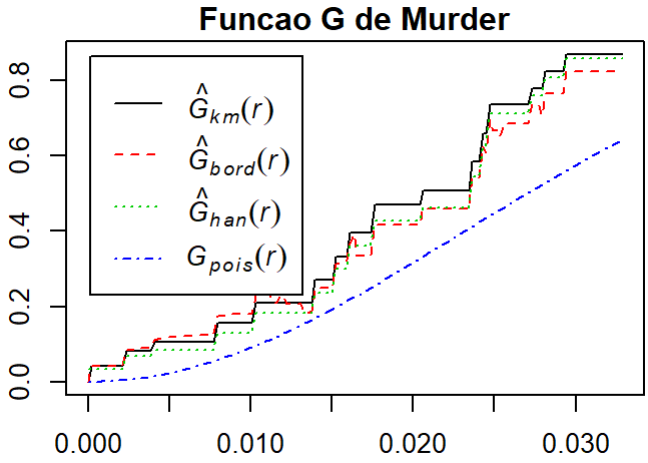

```
Hrob.G=Gest(ahrobbery_ppp)
Hrob.K=Kest(ahrobbery_ppp)
Hrob.F=Fest(ahrobbery_ppp)


par(mfrow = c(2,2))
par(mar=c(2.5,2.5,1.5,.5))
plot(Hrob.G, main="Funcao G de Robbery")
plot(Hrob.K, main="Funcao k de Robbery")
plot(Hrob.F, main="Funcao F de Robbery")
```

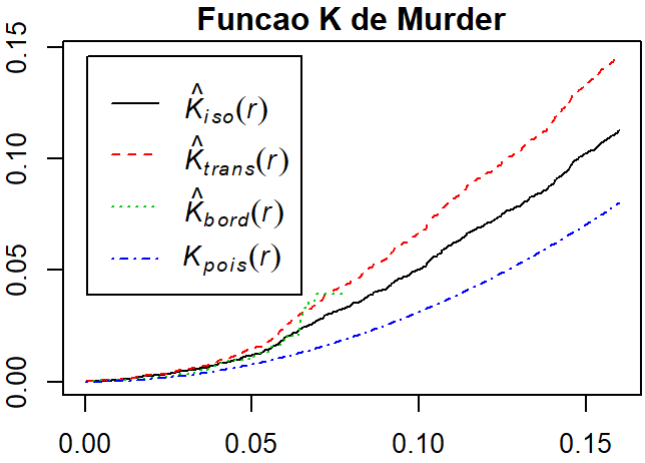


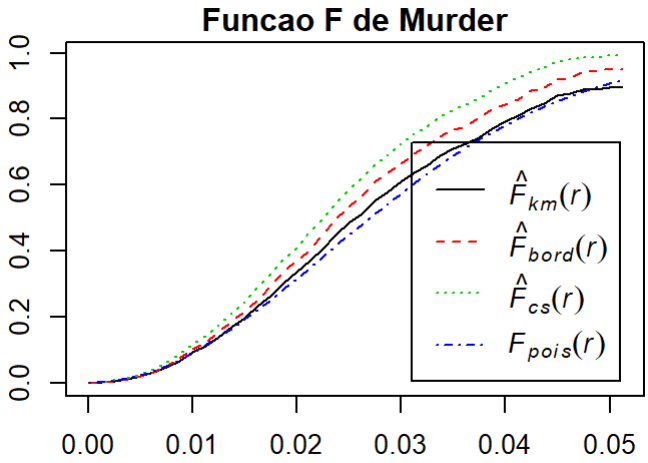




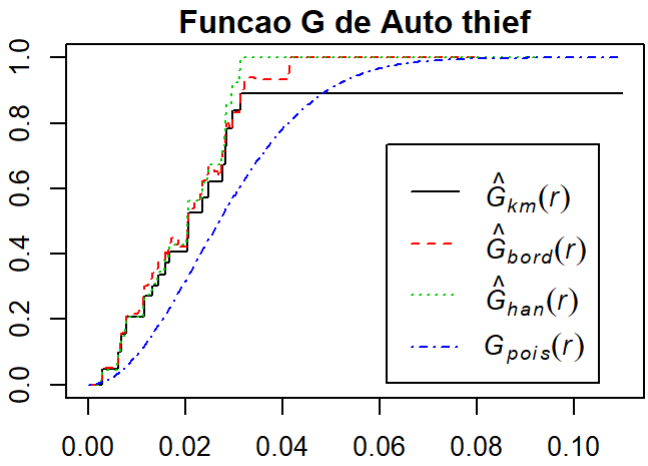

```
Hmurd.G=Gest(ahmurder_ppp)
Hmurd.K=Kest(ahmurder_ppp)
Hmurd.F=Fest(ahmurder_ppp)


par(mfrow = c(2,2))
par(mar=c(2.5,2.5,1.5,.5))

plot(Hmurd.G, main="Funcao G de Murder")
plot(Hmurd.K, main="Funcao K de Murder")
plot(Hmurd.F, main="Funcao F de Murder")
```

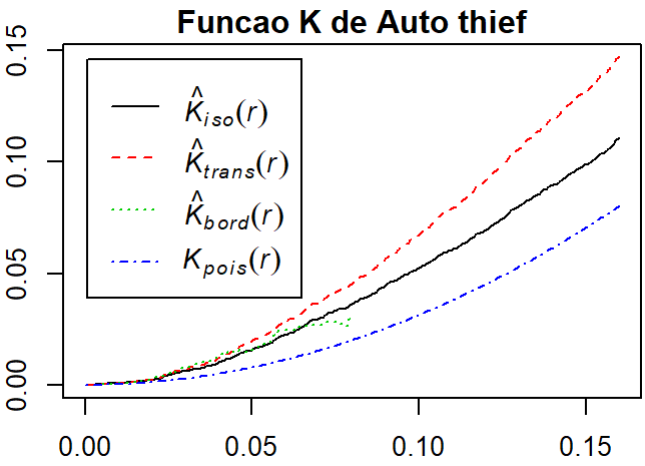


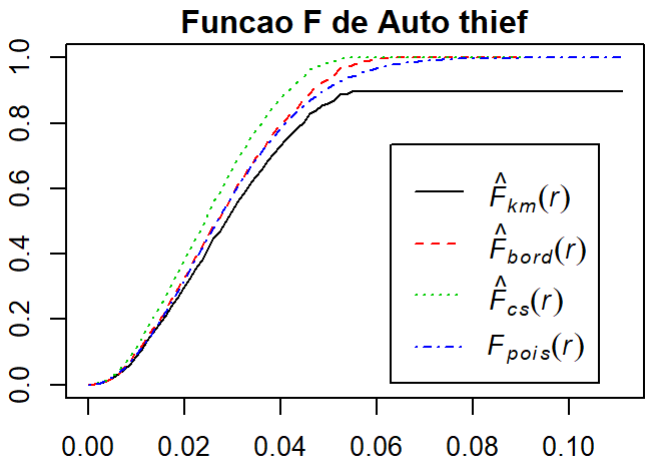




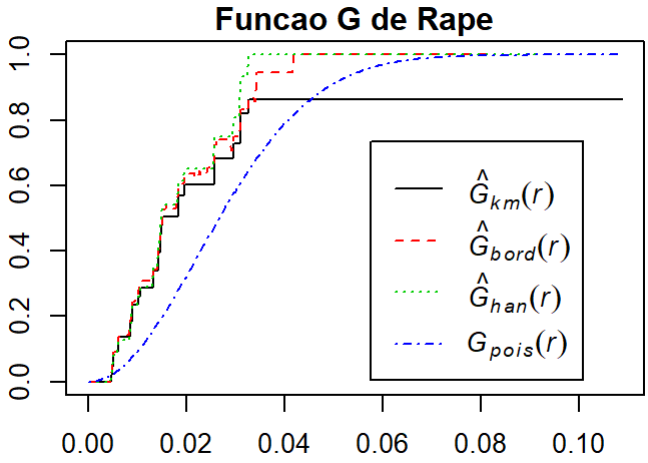

```
Haut.G =Gest(ahauto_ppp)
Haut.K =Kest(ahauto_ppp)
Haut.F =Fest(ahauto_ppp)


par(mfrow = c(2,2))
par(mar=c(2.5,2.5,1.5,.5))

plot(Haut.G, main="Funcao G de Auto thief")
plot(Haut.K, main="Funcao K de Auto thief")
plot(Haut.F, main="Funcao F de Auto thief")
```

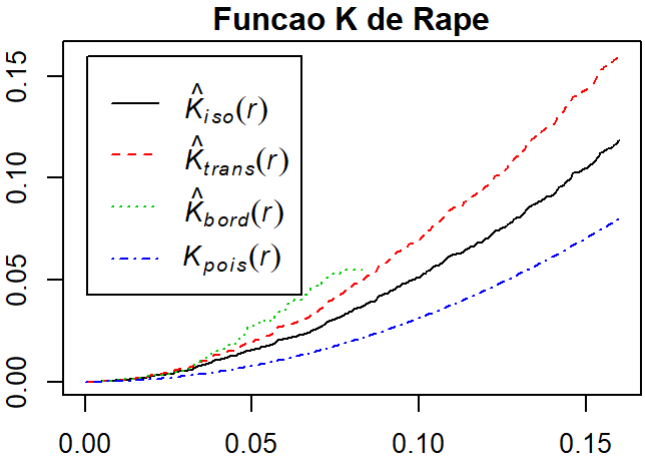


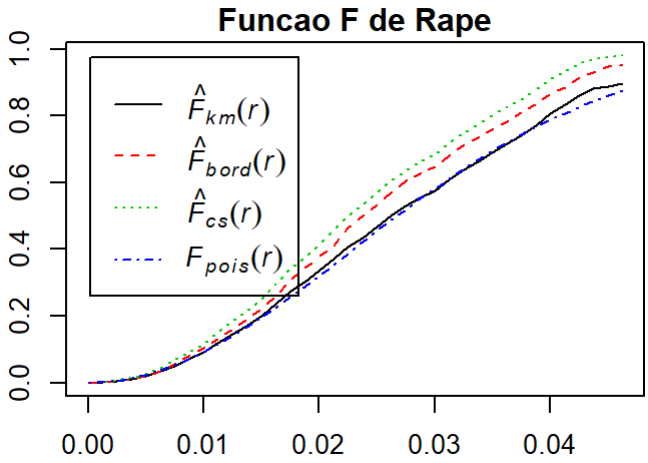




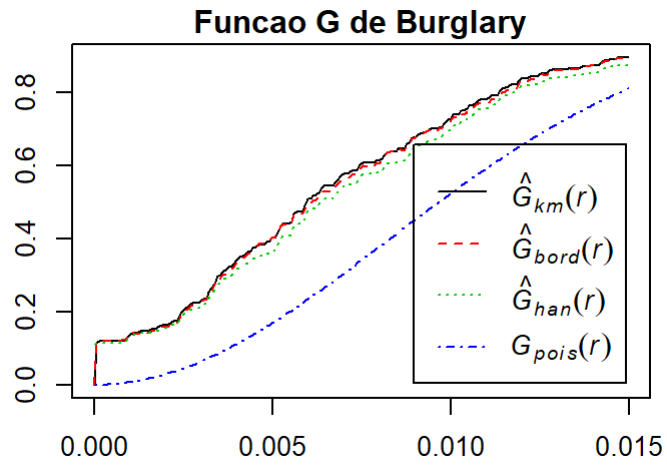

```
Hrape.G= Gest(ahrape_ppp)
Hrape.K=Kest(ahrape_ppp)
Hrape.F=Fest(ahrape_ppp)


par(mfrow = c(2,2))
par(mar=c(2.5,2.5,1.5,.5))

plot(Hrape.G, main="Funcao G de Rape")
plot(Hrape.K, main="Funcao K de Rape")
plot(Hrape.F, main="Funcao F de Rape")
```

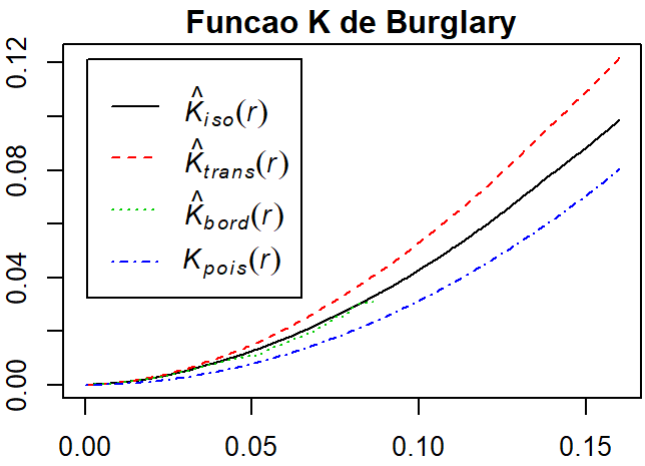


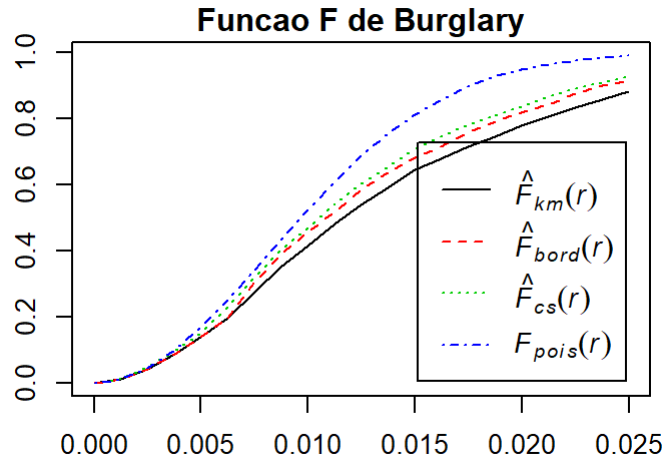





```
Hburg.G=Gest(hburglary_ppp)
Hburg.K=Kest(hburglary_ppp)
Hburg.F=Fest(hburglary_ppp)


par(mfrow = c(2,2))
par(mar=c(2.5,2.5,1.5,.5))
plot(Hburg.G, main="Funcao G de Burglary")
plot(Hburg.K, main="Funcao K de Burglary")
plot(Hburg.F, main="Funcao F de Burglary")
```







Testes

Ao avaliar os testes, Elark-Evans e Hopkins-Skellam podemos descartar hipotese de CSR, ou seja, podemos descartar hipotese de completa aleatoriedade espacial.

Realizando o teste de Clark-Evans para verificar agregacao espacial

```
#-------------------------------------------------------------------------------------------
#Realizando o teste de Clark-Evans para verificar agregacao espacial
clarkevans.test(aHouston_ppp)
```

```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
##
## data:  aHouston_ppp
## R = 0.70754, p-value = 7.613e-05
## alternative hypothesis: two-sided
```

```
clarkevans.test(ahagg_ppp)
```

```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
##
## data:  ahagg_ppp
## R = 0.85166, p-value = 0.04698
## alternative hypothesis: two-sided
```

```
clarkevans.test(ahrobbery_ppp)
```

```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
##
## data:  ahrobbery_ppp
## R = 0.87879, p-value = 0.1046
## alternative hypothesis: two-sided
```

```
clarkevans.test(ahmurder_ppp)
```

```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
##
## data:  ahmurder_ppp
## R = 0.86342, p-value = 0.0674
## alternative hypothesis: two-sided
```

```
clarkevans.test(ahauto_ppp)
```

```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
##
## data:  ahauto_ppp
## R = 0.87947, p-value = 0.1065
## alternative hypothesis: two-sided
```

```
clarkevans.test(ahrape_ppp)
```

```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
##
## data:  ahrape_ppp
## R = 0.90712, p-value = 0.209
## alternative hypothesis: two-sided
```

```
clarkevans.test(hburglary_ppp )
```

```
##
##  Clark-Evans test
##  No edge correction
##  Z-test
##
## data:  hburglary_ppp
## R = 0.76614, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Realizando o teste de Hopkins-Skellam de Completa aleatoriedade espacial para verificar agregacao espacial

```
#-------------------------------------------------------------------------------------------
----
#Realizando o teste de Hopkins-Skellam de Completa aleatoriedade espacial para verificar agre
gacao espacial
hopskel.test(aHouston_ppp, alternative="clustered")
```

```
##
##  Hopkins-Skellam test of CSR
##  using F distribution
##
## data:  aHouston_ppp
## A = 0.1994, p-value = 9.187e-15
## alternative hypothesis: clustered (A < 1)
```

```
hopskel.test(ahagg_ppp, alternative="clustered")
```

```
##
##  Hopkins-Skellam test of CSR
##  using F distribution
##
## data:  ahagg_ppp
## A = 0.16405, p-value < 2.2e-16
## alternative hypothesis: clustered (A < 1)
```

```
hopskel.test(ahrobbery_ppp, alternative="clustered")
```

```
##
##  Hopkins-Skellam test of CSR
##  using F distribution
##
## data:  ahrobbery_ppp
## A = 0.23878, p-value = 5.225e-12
## alternative hypothesis: clustered (A < 1)
```

```
hopskel.test(ahmurder_ppp, alternative="clustered")
```

```
##
##  Hopkins-Skellam test of CSR
##  using F distribution
##
## data:  ahmurder_ppp
## A = 0.16416, p-value < 2.2e-16
## alternative hypothesis: clustered (A < 1)
```

```
hopskel.test(ahauto_ppp, alternative="clustered")
```

```
##
##  Hopkins-Skellam test of CSR
##  using F distribution
##
## data:  ahauto_ppp
## A = 0.15883, p-value < 2.2e-16
## alternative hypothesis: clustered (A < 1)
```

```
hopskel.test(ahrape_ppp, alternative="clustered")
```

```
##
##  Hopkins-Skellam test of CSR
##  using F distribution
##
## data:  ahrape_ppp
## A = 0.099143, p-value < 2.2e-16
## alternative hypothesis: clustered (A < 1)
```

```
hopskel.test(hburglary_ppp , alternative="clustered")
```

```
##
##  Hopkins-Skellam test of CSR
##  using F distribution
##
## data:  hburglary_ppp
## A = 0.074685, p-value < 2.2e-16
## alternative hypothesis: clustered (A < 1)
```

- Kiso -Estimativa de correção isotrópica de K
- Ktrans-estimativa corrigida para tradução de K
- Kbord-estimativa corrigida para as fronteiras de K

- Kpois- Posição teórica de K

Para essa análise usaremos o método de vizinho mais próximo, onde estimamos a função de distribuição cumulativa baseado nas distâncias entre eventos em uma região de análise. Usaremos a hipótese de CSR para estimar a aleatoriedade espacial. As funções observada estão dentro do envelope, o que indica que os dados explorados seguem o CSR em todas as distâncias.

```
#Funcoes para estimar os envelopes das funcoes F, G e K
#Kest=envelope(aNYppp,Kest,nsim=10) #alto custo computacional

par(mfrow=c(2,2))
aHouston_ppp.Gest=envelope(aHouston_ppp,fun = Gest, nsim=20)
```

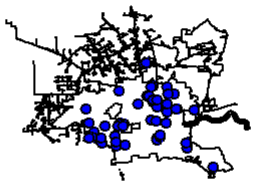
```
## Generating 20 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  20.
##
## Done.
```

```
aHouston_ppp.Fest=envelope(aHouston_ppp,fun = Fest, nsim=10)
```

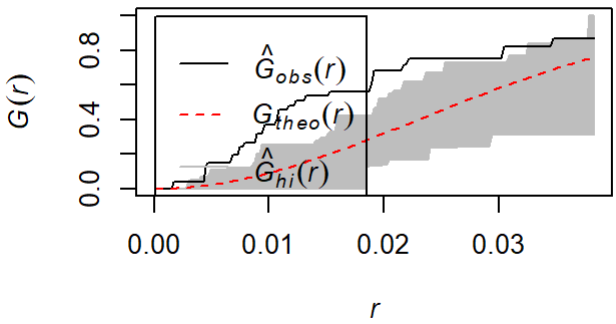
```
## Generating 10 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9,  10.
##
## Done.
```

```
plot(aHouston_ppp, pch=21, cex=0.9, bg="blue")
plot(aHouston_ppp.Gest)
plot(aHouston_ppp.Fest)
```

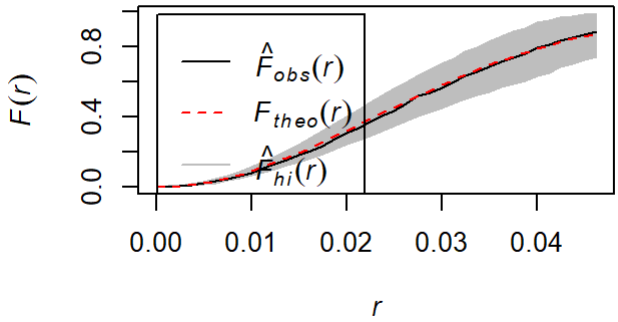
aHouston_ppp



aHouston_ppp.Gest



aHouston_ppp.Fest




```
#Funcoes para estimar os envelopes das funcoes F, G e K
#Kest=envelope(aNYppp,Kest,nsim=10) #alto custo computacional

par(mfrow=c(2,2))
ahagg_ppp.Gest=envelope(ahagg_ppp,fun = Gest,nsim=20)
```

```
## Generating 20 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  20.
##
## Done.
```

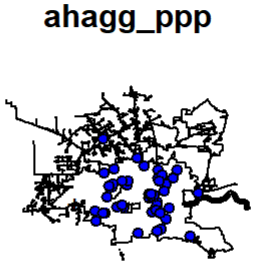
```
ahagg_ppp.Fest=envelope(ahagg_ppp,fun = Fest, nsim=10)
```

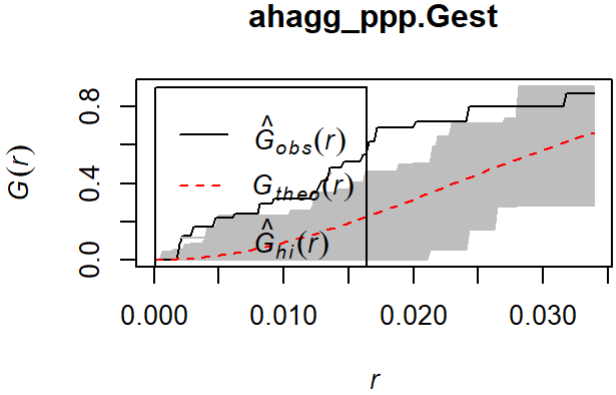
```
## Generating 10 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9,  10.
##
## Done.
```

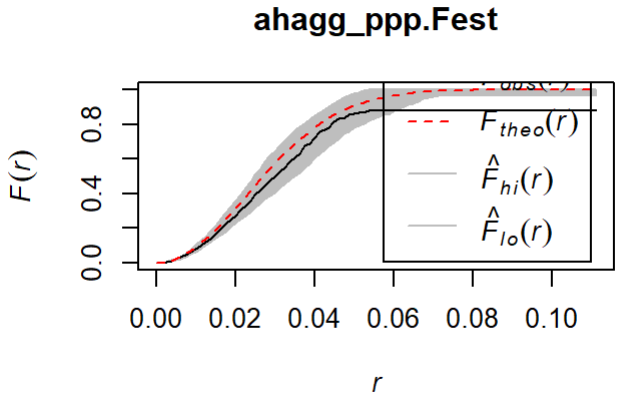
```
plot(ahagg_ppp, pch=21, cex=0.9, bg="blue")
```

```
## Warning in plot.ppp(ahagg_ppp, pch = 21, cex = 0.9, bg = "blue"): 1 illegal
## points also plotted
```

```
plot(ahagg_ppp.Gest)
plot(ahagg_ppp.Fest)
```








```
#Funcoes para estimar os envelopes das funcoes F, G e K
#Kest=envelope(aNYppp,Kest,nsim=10) #alto custo computacional

par(mfrow=c(2,2))
ahmurder_ppp.Gest=envelope(ahmurder_ppp,fun = Gest,nsim=20)
```

```
## Generating 20 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  20.
##
## Done.
```

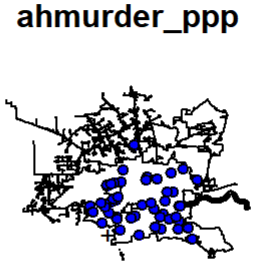
```
ahmurder_ppp.Fest=envelope(ahmurder_ppp,fun = Fest, nsim=10)
```

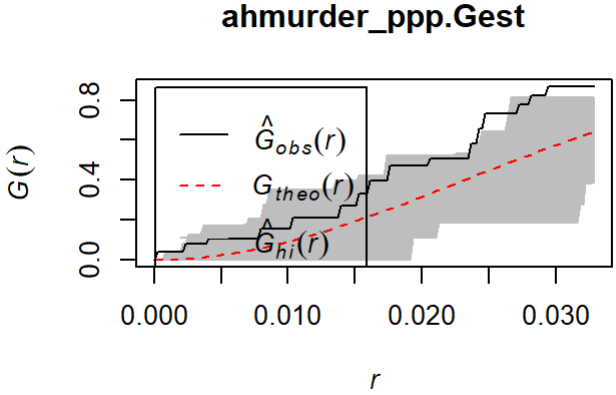
```
## Generating 10 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9,  10.
##
## Done.
```

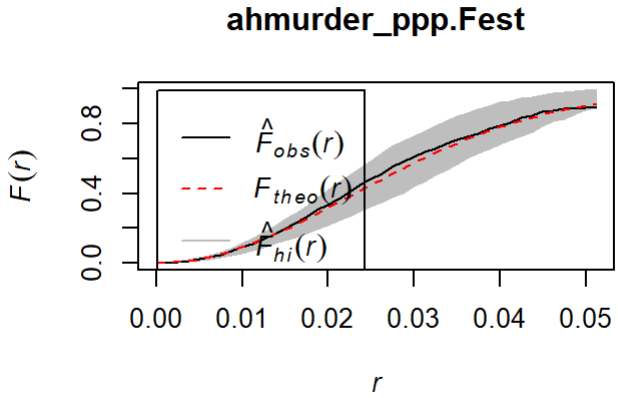
```
plot(ahmurder_ppp, pch=21, cex=0.9, bg="blue")
```

```
## Warning in plot.ppp(ahmurder_ppp, pch = 21, cex = 0.9, bg = "blue"): 1
## illegal points also plotted
```

```
plot(ahmurder_ppp.Gest)
plot(ahmurder_ppp.Fest)
```








```
#Funcoes para estimar os envelopes das funcoes F, G e K
#Kest=envelope(aNYppp,Kest,nsim=10) #alto custo computacional


par(mfrow=c(2,2))
ahrobbery_ppp.Gest=envelope(ahrobbery_ppp,fun = Gest,nsim=20)
```

```
## Generating 20 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  20.
##
## Done.
```

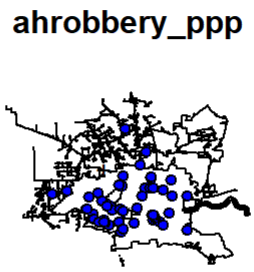
```
ahrobbery_ppp.Fest=envelope(ahrobbery_ppp,fun = Fest, nsim=10)
```

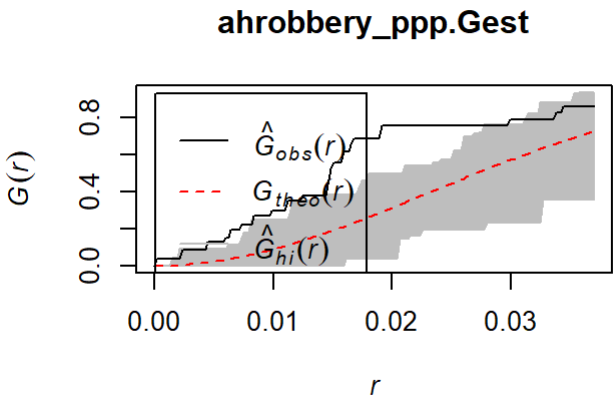
```
## Generating 10 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9,  10.
##
## Done.
```

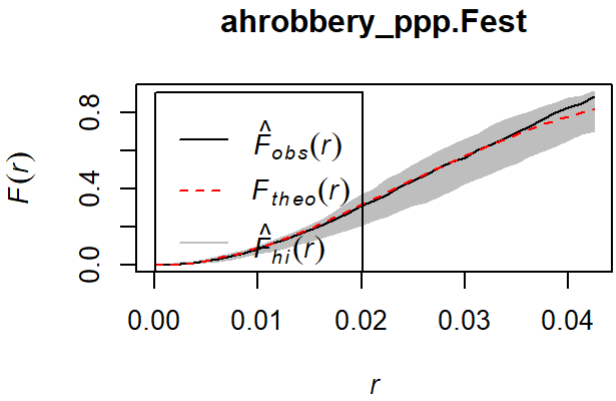
```
plot(ahrobbery_ppp, pch=21, cex=0.9, bg="blue")
```

```
## Warning in plot.ppp(ahrobbery_ppp, pch = 21, cex = 0.9, bg = "blue"): 1
## illegal points also plotted
```

```
plot(ahrobbery_ppp.Gest)
plot(ahrobbery_ppp.Fest)
```








```
#Funcoes para estimar os envelopes das funcoes F, G e K
#Kest=envelope(aNYppp,Kest,nsim=10) #alto custo computacional
par(mfrow=c(2,2))
hburglary_ppp.Gest=envelope(hburglary_ppp,fun = Gest,nsim=20)
```

```
## Generating 20 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  20.
##
## Done.
```

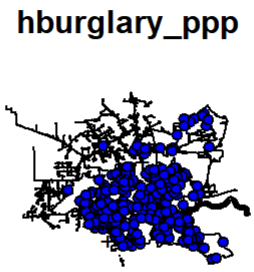
```
hburglary_ppp.Fest=envelope(hburglary_ppp,fun = Fest, nsim=10)
```

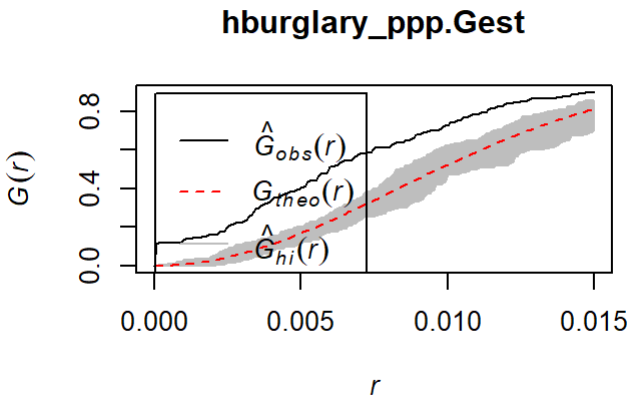
```
## Generating 10 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9,  10.
##
## Done.
```

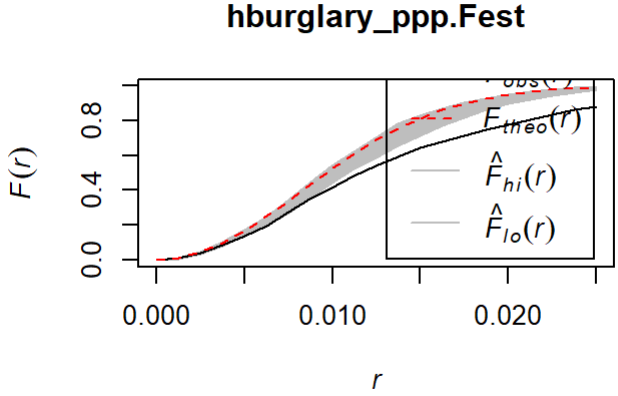
```
plot(hburglary_ppp, pch=21, cex=0.9, bg="blue")
```

```
## Warning in plot.ppp(hburglary_ppp, pch = 21, cex = 0.9, bg = "blue"): 6
## illegal points also plotted
```

```
plot(hburglary_ppp.Gest)
plot(hburglary_ppp.Fest)
```








```
#Funcoes para estimar os envelopes das funcoes F, G e K
#Kest=envelope(aNYppp,Kest,nsim=10) #alto custo computacional

par(mfrow=c(2,2))
ahrape_ppp.Gest=envelope(ahrape_ppp,fun = Gest,nsim=20)
```

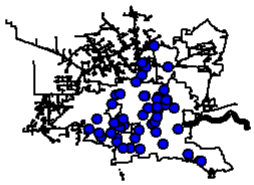
```
## Generating 20 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  20.
##
## Done.
```

```
ahrape_ppp.Fest=envelope(ahrape_ppp,fun = Fest, nsim=10)
```

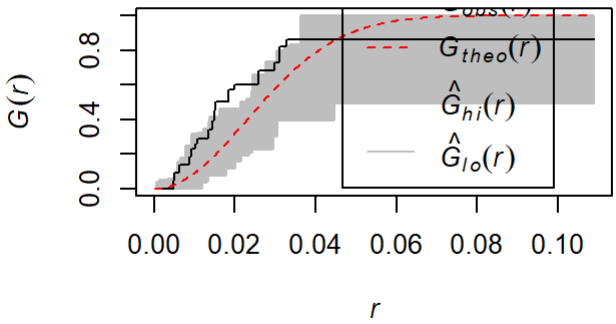
```
## Generating 10 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9,  10.
##
## Done.
```

```
plot(ahrape_ppp, pch=21, cex=0.9, bg="blue")
plot(ahrape_ppp.Gest)
plot(ahrape_ppp.Fest)
```

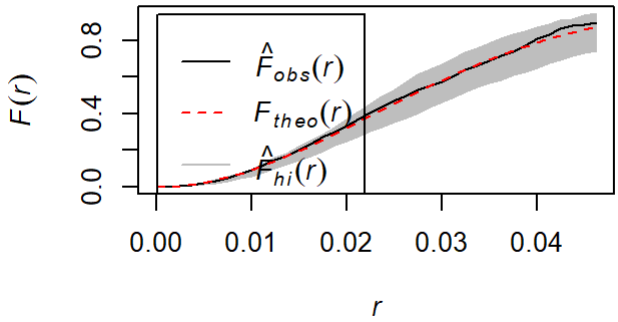
ahrape_ppp



ahrape_ppp.Gest



ahrape_ppp.Fest




```
#Funcoes para estimar os envelopes das funcoes F, G e K
#Kest=envelope(aNYppp,Kest,nsim=10) #alto custo computacional

par(mfrow=c(2,2))
ahauto_ppp.Gest=envelope(ahauto_ppp,fun = Gest,nsim=20)
```

```
## Generating 20 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,  20.
##
## Done.
```

```
ahauto_ppp.Fest=envelope(ahauto_ppp,fun = Fest, nsim=10)
```

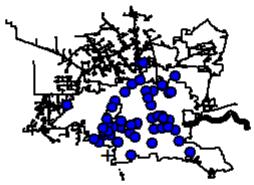
```
## Generating 10 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9,  10.
##
## Done.
```

```
plot(ahauto_ppp, pch=21, cex=0.9, bg="blue")
```

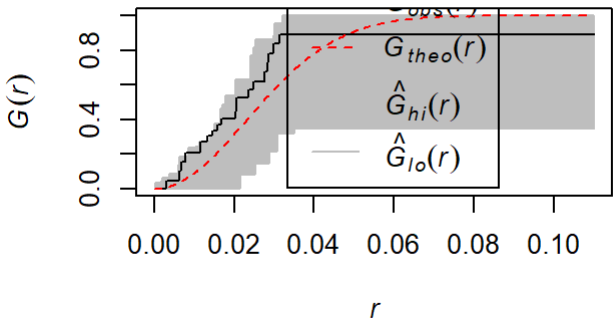
```
## Warning in plot.ppp(ahauto_ppp, pch = 21, cex = 0.9, bg = "blue"): 1
## illegal points also plotted
```

```
plot(ahauto_ppp.Gest)
plot(ahauto_ppp.Fest)
```

ahauto_ppp



ahauto_ppp.Gest



ahauto_ppp.Fest