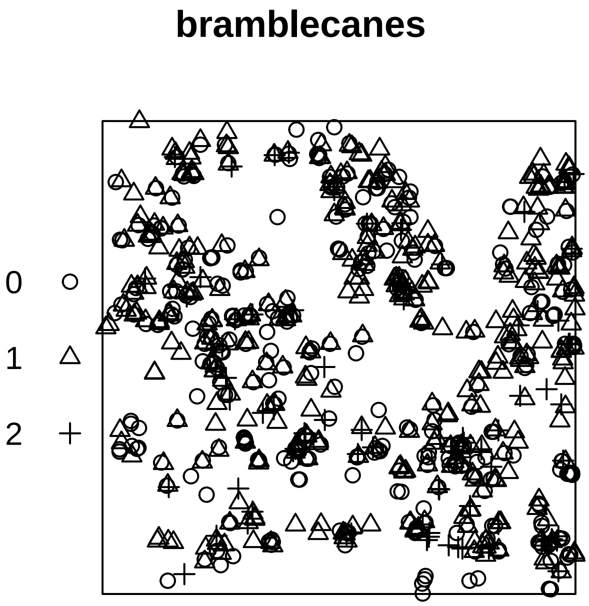
Homework 5

Peter Tran

```
library(grid)
library(oldtmaptools)
library(tidyverse)
library(tmap)
library(sp)
library(spatstat)
```

Let's read in our bramblecane data.

```
data(bramblecanes)
plot(bramblecanes)
```



That is such an *ugly* plot…

We need to separate the bramblecanes into their own groups.

```
bramblecanes_pos = cbind(bramblecanes$x, bramblecanes$y)
bramblecanes_0 = SpatialPoints(bramblecanes_pos[bramblecanes$marks == 0,])
bramblecanes_1 = SpatialPoints(bramblecanes_pos[bramblecanes$marks == 1,])
bramblecanes_2 = SpatialPoints(bramblecanes_pos[bramblecanes$marks == 2,])
```

Our first task is to construct kernal density estimates and display them as heatmaps.

```
bramblecanes_0_density <- smooth_map(bramblecanes_0, bandwidth = c(0.00005, 0.00005))
```

```
##   |                                                                              |
bramblecanes_1_density <- smooth_map(bramblecanes_1, bandwidth = c(0.00005, 0.00005))
```

```
##   |                                                                                 |
```

```
bramblecanes_2_density <- smooth_map(bramblecanes_2, bandwidth = c(0.00005, 0.00005))
```

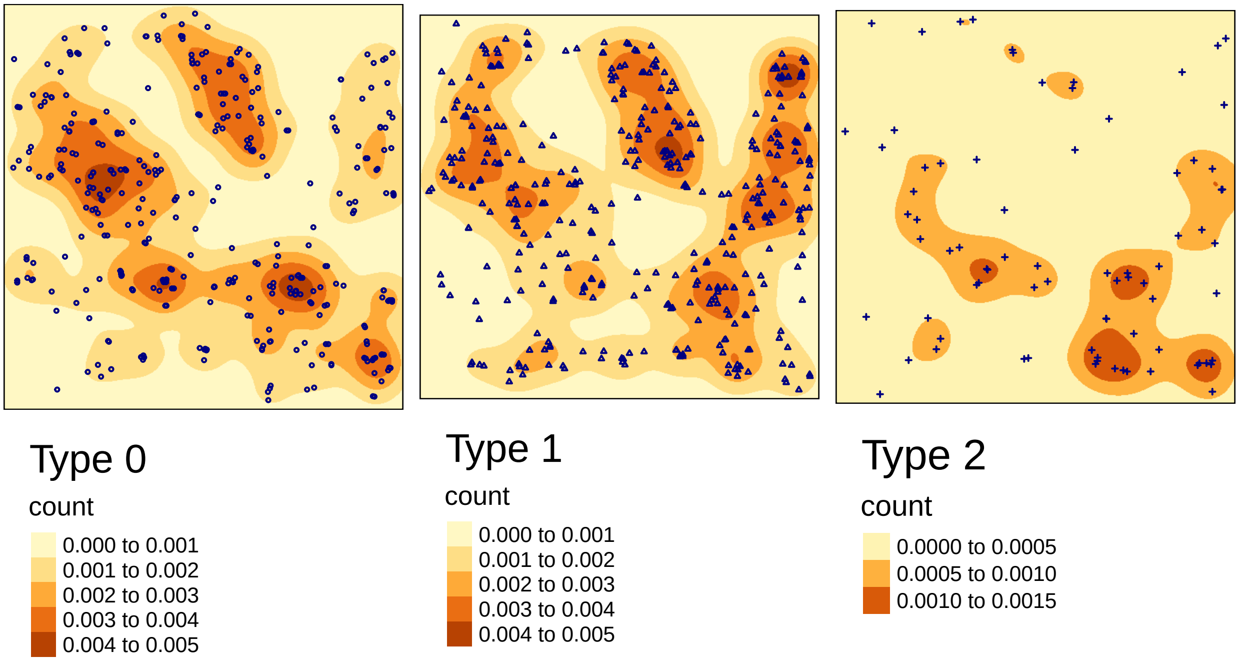
```
##   |                                                                                 |
```

```
bramblecanes_0_heatmap <- tm_shape(bramblecanes_0_density$raster) +
  tm_raster() +
  tm_shape(bramblecanes_0) +
  tm_dots(col = "navyblue", shape = 1) +
  tm_layout(title = "Type 0", legend.outside = T, legend.outside.position = "bottom")

bramblecanes_1_heatmap <- tm_shape(bramblecanes_1_density$raster) +
  tm_raster() +
  tm_shape(bramblecanes_1) +
  tm_dots(col = "navyblue", shape = 2) +
  tm_layout(title = "Type 1", legend.outside = T, legend.outside.position = "bottom")

bramblecanes_2_heatmap <- tm_shape(bramblecanes_2_density$raster) +
  tm_raster() +
  tm_shape(bramblecanes_2) +
  tm_dots(col = "navyblue", shape = 3) +
  tm_layout(title = "Type 2", legend.outside = T, legend.outside.position = "bottom")

grid.newpage()
pushViewport(viewport(layout = grid.layout(1, 3)))
print(bramblecanes_0_heatmap, vp = viewport(layout.pos.col = 1, height = 5))
print(bramblecanes_1_heatmap, vp = viewport(layout.pos.col = 2, height = 5))
print(bramblecanes_2_heatmap, vp = viewport(layout.pos.col = 3, height = 5))
```



From this it appears that there's definitely some clustering happening with type 0 and 1, but not so much 2.

Now we will estimate the K function for each of these types, and then do an envelope analysis to examine for clustering.

```
par(mfrow = c(3, 1))
plot(envelope(subset(bramblecanes, marks == 0), Kest, correction = "border", nsim = 1000),
     main = "Envelope for type 0")
```

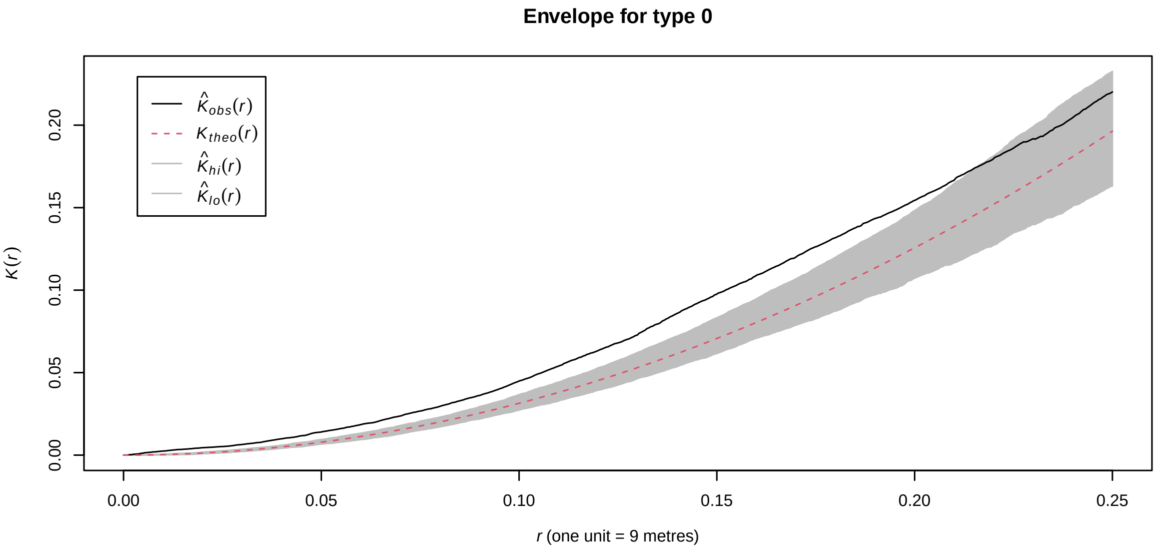
```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```

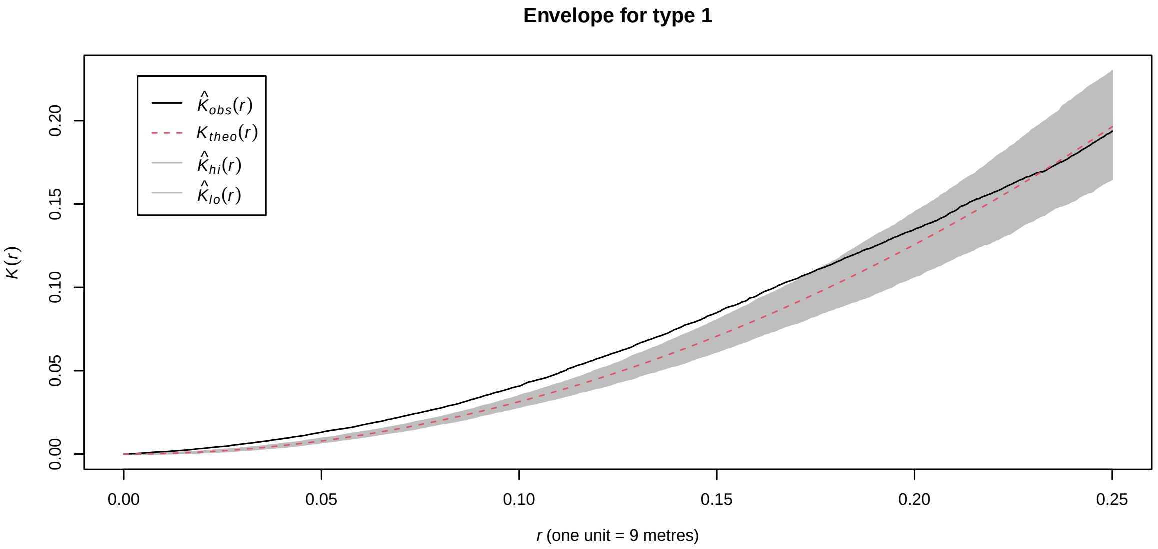
```
plot(envelope(subset(bramblecanes, marks == 1), Kest, correction = "border", nsim = 1000),
     main = "Envelope for type 1")
```

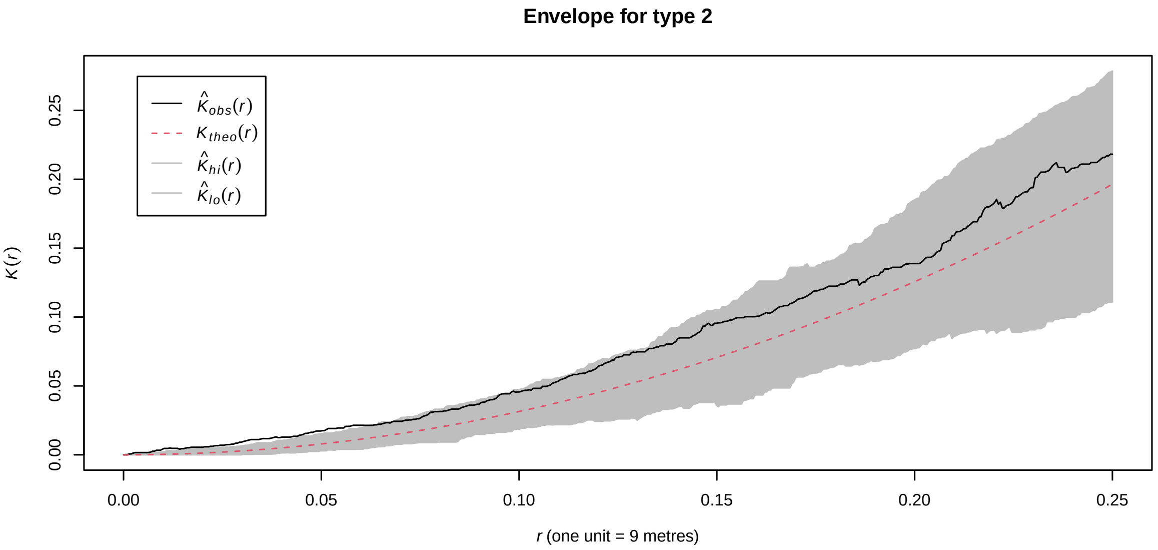
```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```

```
plot(envelope(subset(bramblecanes, marks == 2), Kest, correction = "border", nsim = 1000),
     main = "Envelope for type 2")
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```


Envelope for type 0


Envelope for type 1


Envelope for type 2

From this, we can see that type 0 bramblecanes definitely exhibit clustering, as the K estimate is higher than what K would be under the CSR assumption. It does fall into the envelope as the radius r becomes large, however. Type 1 seems to also exhibit some clustering, but it goes under the envelope quicker than type 0 does. Type 2 starts out as higher than the envelope, but it stays too close to the envelope and quickly falls under to be comfortable saying that it is showing clustering, especially if you look at the heatmap produced from the KDE above.

Let’s do the same thing, but now with the G function.

```r
par(mfrow = c(3, 1))
plot(envelope(subset(bramblecanes, marks == 0), Gest, correction = "border", nsim = 1000),
     main = "Envelope for type 0")
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```

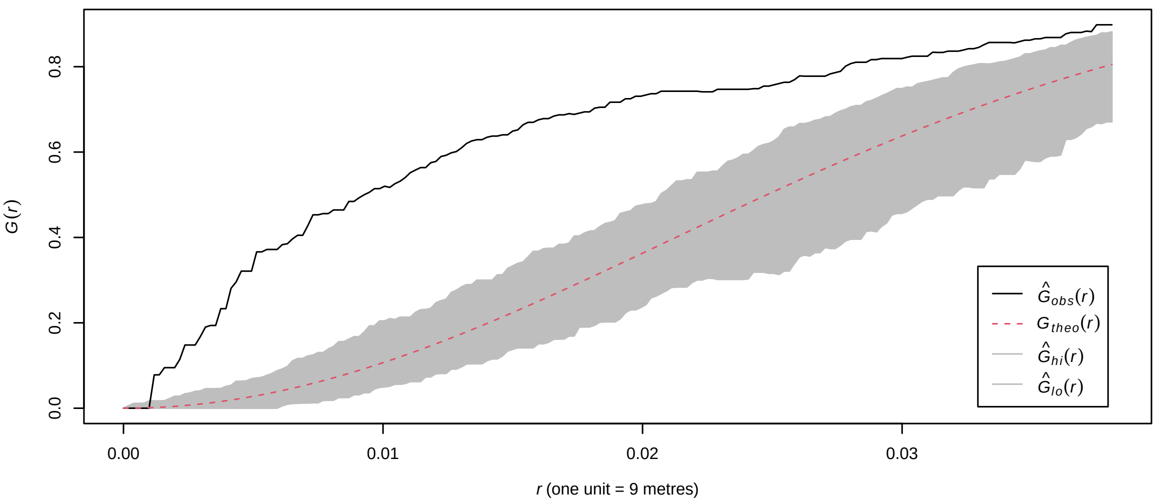
```r
plot(envelope(subset(bramblecanes, marks == 1), Gest, correction = "border", nsim = 1000),
     main = "Envelope for type 1")
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```

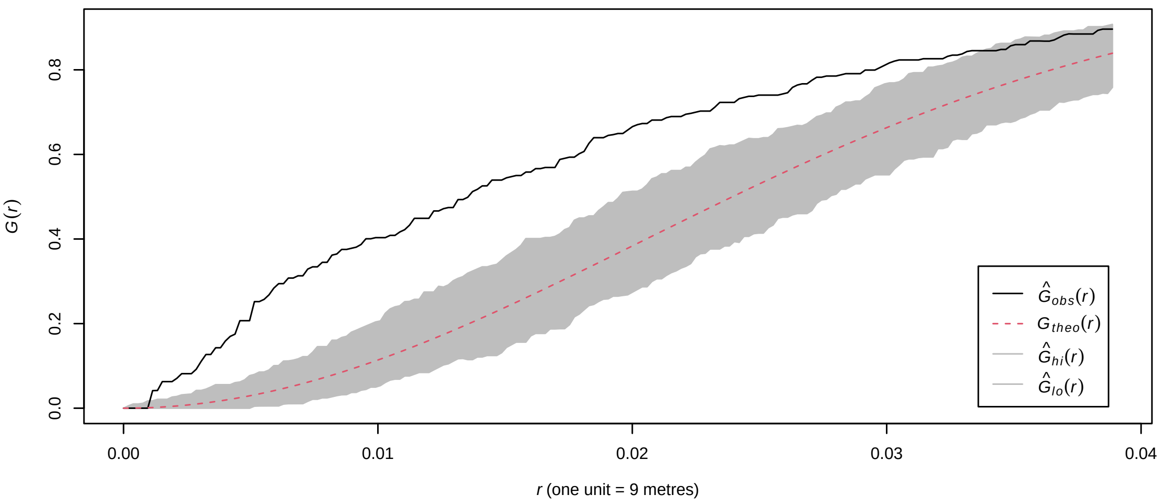
```r
plot(envelope(subset(bramblecanes, marks == 2), Gest, correction = "border", nsim = 1000),
     main = "Envelope for type 2")
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```

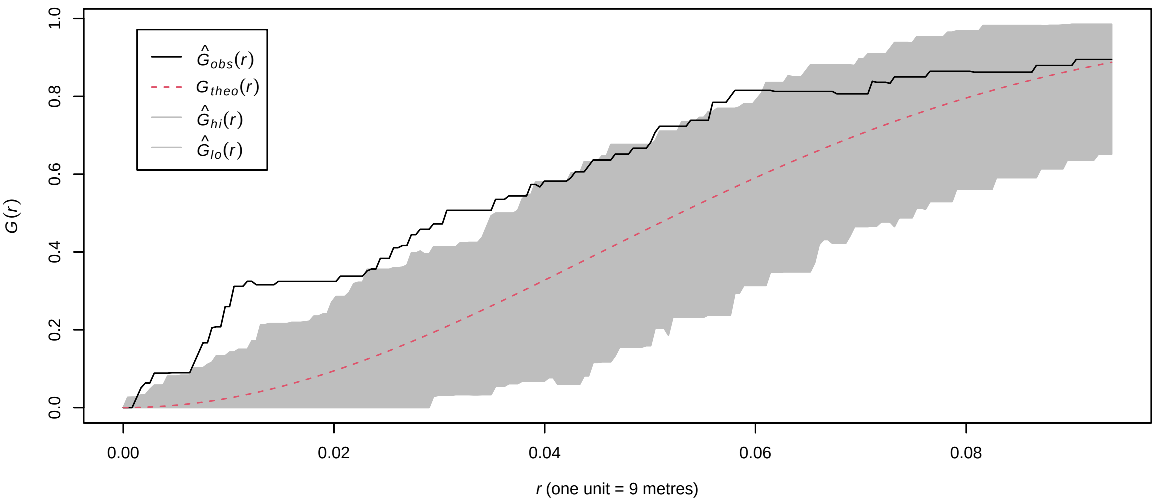
Envelope for type 0



Envelope for type 1



Envelope for type 2



Using the G function, we see a similar pattern to what we saw with the K function. The main difference here is that type 1 looks like it has stronger evidence for exhibiting clustering.

Let's now look at using the mad and dclf test as an omnibus test for each of these. I'll use the K function because I'm most comfortable with my understanding of that function.

```
mad.test(subset(bramblecanes, marks == 0), Kest, verbose = F, nsim = 1000)
```

```
##
##  Maximum absolute deviation test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: K(r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Maximum absolute deviation
##  Deviation = observed minus theoretical
##
## data:  subset(bramblecanes, marks == 0)
## mad = 0.020362, rank = 1, p-value = 0.000999
```

```
dclf.test(subset(bramblecanes, marks == 0), Kest, verbose = F, nsim = 1000)
```

```
##
##  Diggle-Cressie-Loosmore-Ford test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: K(r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Integral of squared absolute deviation
##  Deviation = observed minus theoretical
##
## data:  subset(bramblecanes, marks == 0)
## u = 4.6451e-05, rank = 1, p-value = 0.000999
```

Looks like we have very strong evidence that there is clustering for type 0.

```
mad.test(subset(bramblecanes, marks == 1), Kest, verbose = F, nsim = 1000)
```

```
##
##  Maximum absolute deviation test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: K(r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Maximum absolute deviation
##  Deviation = observed minus theoretical
##
## data:  subset(bramblecanes, marks == 1)
## mad = 0.017229, rank = 1, p-value = 0.000999
```

```
dclf.test(subset(bramblecanes, marks == 1), Kest, verbose = F, nsim = 1000)
```

```
##
##  Diggle-Cressie-Loosmore-Ford test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: K(r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Integral of squared absolute deviation
##  Deviation = observed minus theoretical
##
## data:  subset(bramblecanes, marks == 1)
## u = 3.9217e-05, rank = 1, p-value = 0.000999
```

We also have very strong evidence that type 1 exhibits clustering too.

```r
mad.test(subset(bramblecanes, marks == 2), Kest, verbose = F, nsim = 1000)
```

```
##
##  Maximum absolute deviation test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: K(r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Maximum absolute deviation
##  Deviation = observed minus theoretical
##
## data:  subset(bramblecanes, marks == 2)
## mad = 0.053691, rank = 1, p-value = 0.000999
```

```r
dclf.test(subset(bramblecanes, marks == 2), Kest, verbose = F, nsim = 1000)
```

```
##
##  Diggle-Cressie-Loosmore-Ford test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: K(r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Integral of squared absolute deviation
##  Deviation = observed minus theoretical
##
## data:  subset(bramblecanes, marks == 2)
## u = 0.00016684, rank = 1, p-value = 0.000999
```

Looks like we also have strong evidence for type 2 exhibiting clustering. I guess I can believe that, though it's definitely not as strong as the other types.

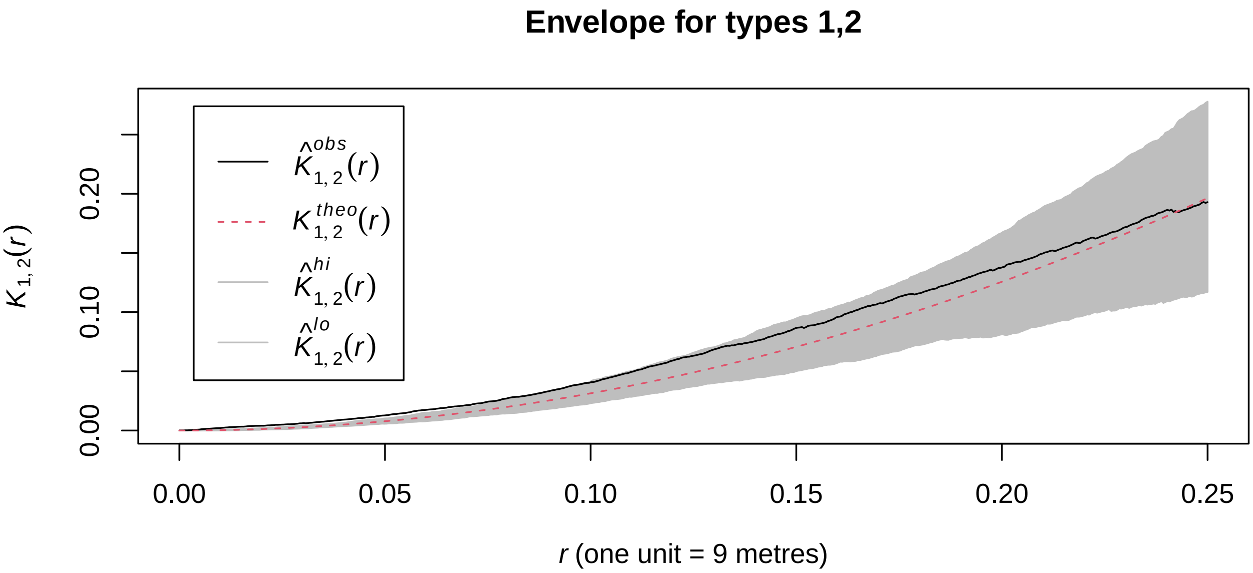
Now we're going to examine the cross K function between type 1 and 2. We'll do it both ways.

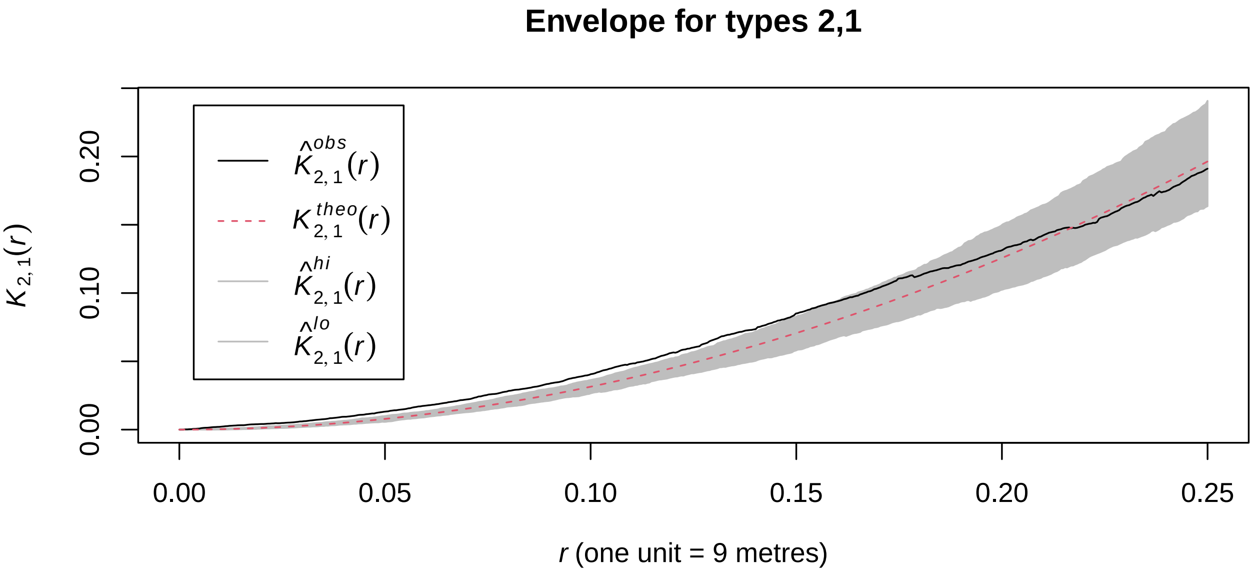
```r
par(mfrow = c(2, 1))
plot(envelope(bramblecanes, Kcross, i = 1, j = 2, correction = "border", nsim = 1000),
     main = "Envelope for types 1,2")
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```

```
plot(envelope(bramblecanes, Kcross, i = 2, j = 1, correction = "border", nsim = 1000),
     main = "Envelope for types 2,1")
```

```
## Generating 1000 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60........
## .70.........80.........90.........100.........110.........120.........130......
## ...140.........150.........160.........170.........180.........190.........200....
## .....210.........220.........230.........240.........250.........260.........270..
## .......280.........290.........300.........310.........320.........330.........340
## .........350.........360.........370.........380.........390.........400........
## .410.........420.........430.........440.........450.........460.........470......
## ...480.........490.........500.........510.........520.........530.........540....
## .....550.........560.........570.........580.........590.........600.........610..
## .......620.........630.........640.........650.........660.........670.........680
## .........690.........700.........710.........720.........730.........740........
## .750.........760.........770.........780.........790.........800.........810......
## ...820.........830.........840.........850.........860.........870.........880....
## .....890.........900.........910.........920.........930.........940.........950..
## .......960.........970.........980.........990......... 1000.
##
## Done.
```





Ehh... It's kind of tough to see here. It looks like 1 tends to be clustered with 2 more than 2 tends to be with 1. Let's also run mad tests just to see.

```
mad.test(bramblecanes, Kcross, i = 1, j = 2, verbose = F, nsim = 1000)
```

```
##
##  Maximum absolute deviation test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: "K"[1, 2](r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Maximum absolute deviation
##  Deviation = observed minus theoretical
##
## data:  bramblecanes
## mad = 0.01364, rank = 7, p-value = 0.006993
```

```
mad.test(bramblecanes, Kcross, i = 2, j = 1, verbose = F, nsim = 1000)
```

```
##
##  Maximum absolute deviation test of CSR
##  Monte Carlo test based on 1000 simulations
##  Summary function: "K"[2, 1](r)
##  Reference function: theoretical
##  Alternative: two.sided
##  Interval of distance values: [0, 0.25] units (one unit = 9 metres)
##  Test statistic: Maximum absolute deviation
##  Deviation = observed minus theoretical
##
## data:  bramblecanes
## mad = 0.016609, rank = 1, p-value = 0.000999
```

Ok, the mad test says that they cluster together in both directions. Though the evidence is much, much stronger in the case of type 1's being close to 2's.

These both seem to be the opposite of what I expect. Looking at the plot of the points above, since type 2 has so fewer points than type 1, I would expect us to see that type 2's tend to be near type 1's and not the other way around. Both the plot and the omnibus test seem to not agree with my manual observation though.