

Population and Community Ecology Homework 2

Samantha Apgar & Henry Frye

Part 1

Tutorial Run-Through

```
# install.packages("spatstat")
library(spatstat)

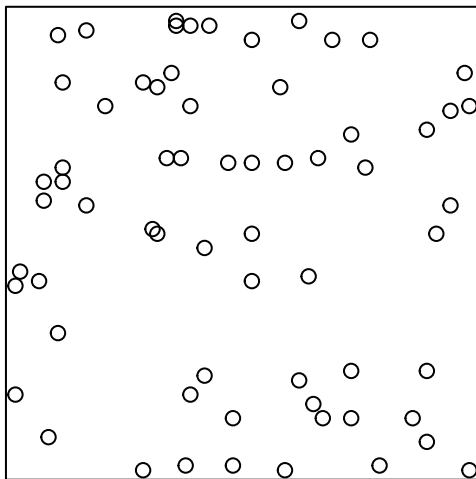
## Loading required package: nlme
## Loading required package: rpart
##
## spatstat 1.49-0      (nickname: 'So-Called Software')
## For an introduction to spatstat, type 'beginner'

data(japanesepines)
summary(japanesepines)

## Planar point pattern: 65 points
## Average intensity 65 points per square unit (one unit = 5.7 metres)
##
## Coordinates are given to 2 decimal places
## i.e. rounded to the nearest multiple of 0.01 units (one unit = 5.7 metres)
##
## Window: rectangle = [0, 1] x [0, 1] units
## Window area = 1 square unit
## Unit of length: 5.7 metres

plot(japanesepines)
```

japanesepines



```
clarkevans(japanesepines)
```

```
##      naive Donnelly      cdf
## 1.064002 1.007507 1.056241
```

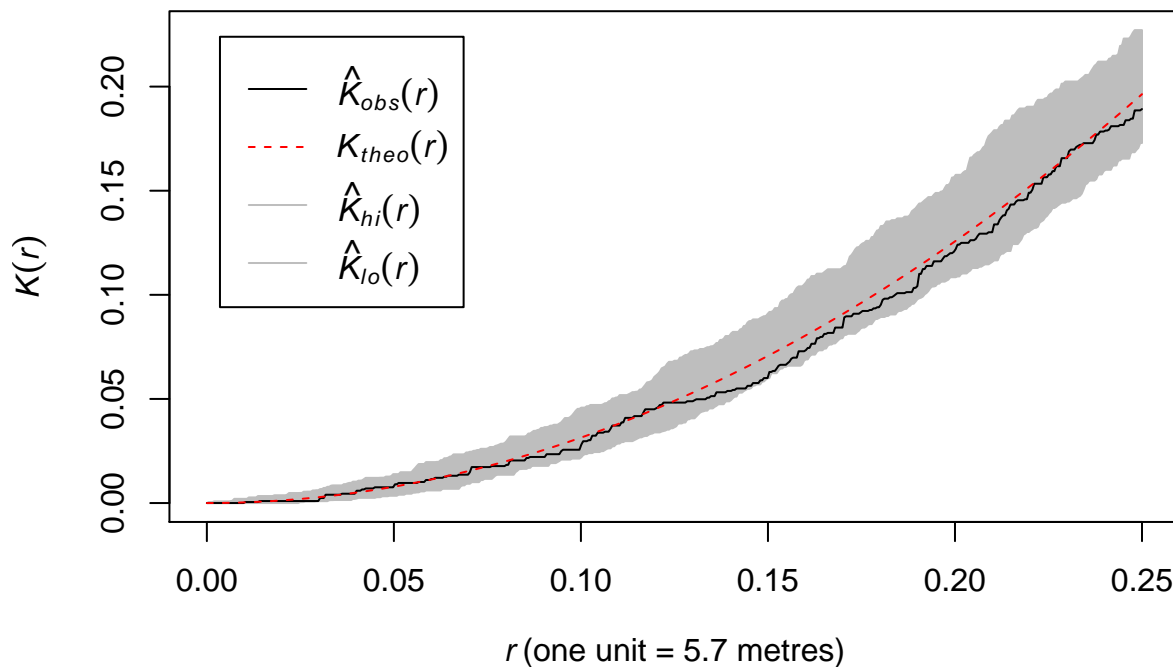
```
clarkevans.test(japanesepines, correction = "Donnelly")
```

```
##
## Clark-Evans test
## Donnelly correction
## Monte Carlo test based on 999 simulations of CSR with fixed n
##
## data: japanesepines
## R = 1.0075, p-value = 0.974
## alternative hypothesis: two-sided
```

```
RipleysK <- Kest(japanesepines)
plot(envelope(japanesepines))
```

```
## Generating 99 simulations of CSR ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.
##
## Done.
```

envelope(japanesepines)

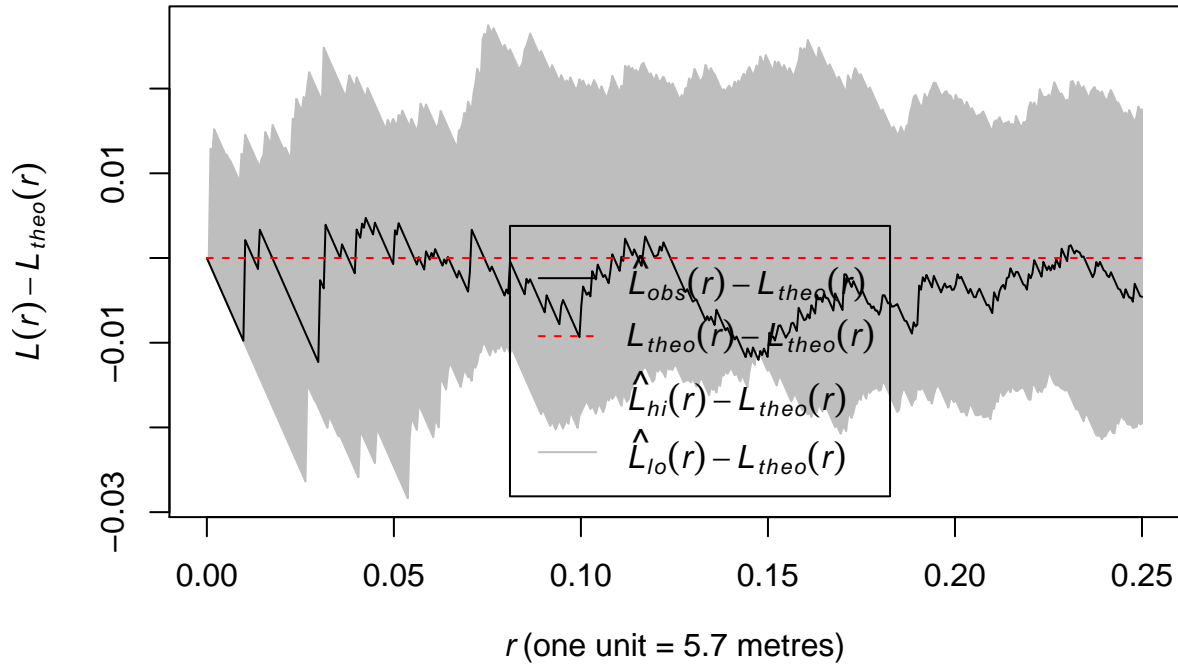


```
RipleysL <- Lest(japanesepines)
plot(envelope(japanesepines, Lest, nsim = 99), .-theo~r)
```

```
## Generating 99 simulations of CSR ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.
##
```

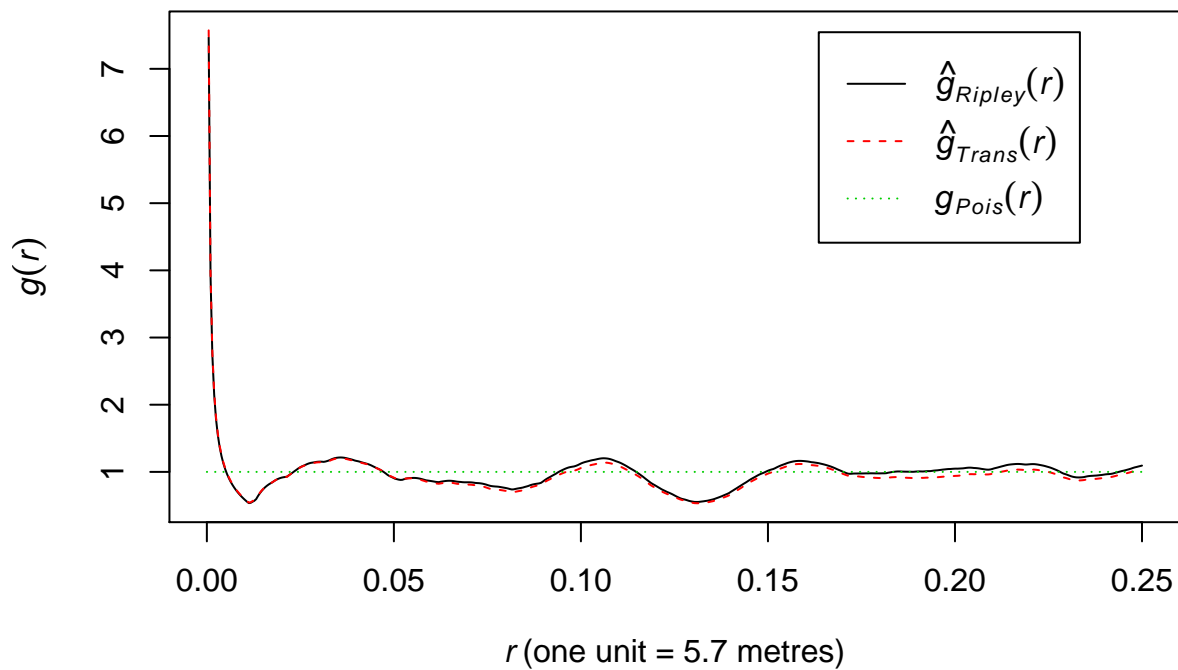
```
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.
##
## Done.
```

envelope(japanesepines, Lest, nsim = 99)



```
p <- pcf(japanesepines)
plot(p)
```

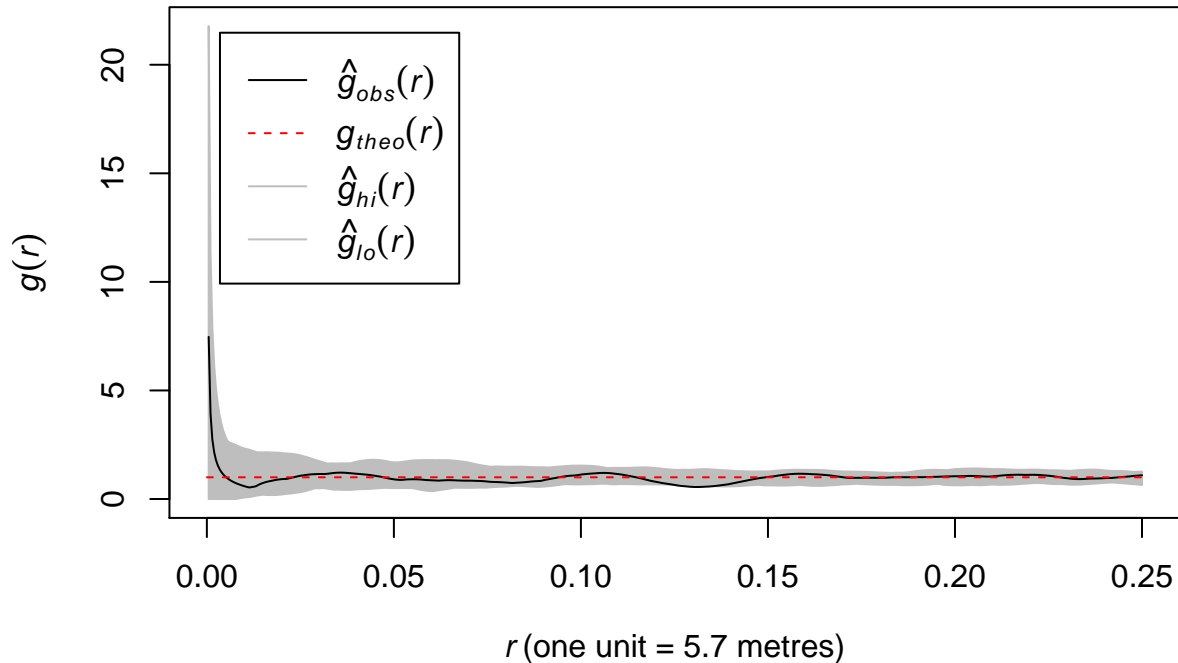
p



```
plot(envelope(japanesepines, pcf, nsim = 99))
```

```
## Generating 99 simulations of CSR ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.
##
## Done.
```

envelope(japanesepines, pcf, nsim = 99)



```
data(longleaf)
summary(longleaf)
```

```
## Marked planar point pattern: 584 points
## Average intensity 0.0146 points per square metre
##
## Coordinates are given to 1 decimal place
## i.e. rounded to the nearest multiple of 0.1 metres
##
## marks are numeric, of type 'double'
## Summary:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      2.00   9.10   26.15   26.84   42.12   75.90
##
## Window: rectangle = [0, 200] x [0, 200] metres
## Window area = 40000 square metres
## Unit of length: 1 metre
```

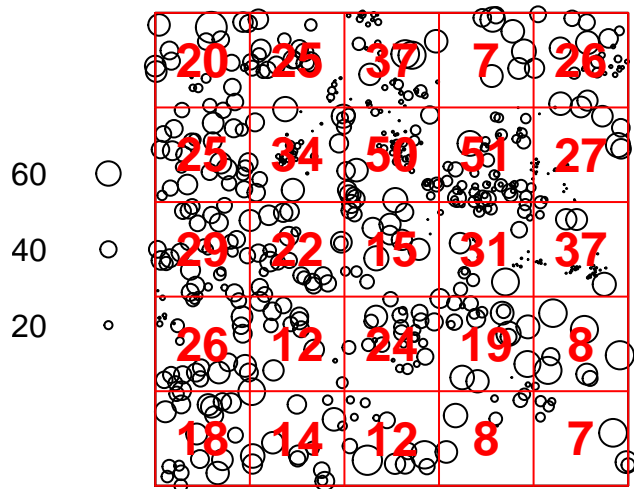
```
QC <- quadratcount(longleaf)
QC
```

```
##           x
```

```
## y      [0,40) [40,80) [80,120) [120,160) [160,200]
## [160,200]    20     25      37       7      26
## [120,160)    25     34      50      51      27
## [80,120)     29     22      15      31      37
## [40,80)      26     12      24      19      8
## [0,40)       18     14      12       8      7
```

```
plot(longleaf)
plot(QC, add = TRUE, cex = 1.5, col = "red", font = 2)
```

longleaf



```
QC <- quadratcount(longleaf, 15, 15)
QC
```

```
## x
## y      [0,13.333333333333333)
## [186.6666666666667,200] 1
## [173.3333333333333,186.6666666666667) 4
## [160,173.3333333333333) 0
## [146.6666666666667,160) 3
## [133.3333333333333,146.6666666666667) 4
## [120,133.3333333333333) 2
## [106.6666666666667,120) 1
## [93.3333333333333,106.6666666666667) 5
## [80,93.3333333333333) 1
## [66.6666666666667,80) 7
## [53.3333333333333,66.6666666666667) 1
## [40,53.3333333333333) 5
## [26.6666666666667,40) 3
## [13.3333333333333,26.6666666666667) 0
## [0,13.3333333333333) 5
## x
## y      [13.3333333333333,26.6666666666667)
## [186.6666666666667,200] 2
## [173.3333333333333,186.6666666666667) 1
## [160,173.3333333333333) 5
## [146.6666666666667,160) 1
```

##	[133.333333333333,146.666666666667)	3
##	[120,133.333333333333)	2
##	[106.666666666667,120)	4
##	[93.333333333333,106.666666666667)	3
##	[80,93.333333333333)	5
##	[66.666666666667,80)	2
##	[53.333333333333,66.666666666667)	0
##	[40,53.333333333333)	3
##	[26.666666666667,40)	3
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	2
##		
##	x	
##	y [26.666666666667,40)	
##	[186.666666666667,200]	3
##	[173.333333333333,186.666666666667)	2
##	[160,173.333333333333)	2
##	[146.666666666667,160)	3
##	[133.333333333333,146.666666666667)	4
##	[120,133.333333333333)	3
##	[106.666666666667,120)	2
##	[93.333333333333,106.666666666667)	4
##	[80,93.333333333333)	4
##	[66.666666666667,80)	3
##	[53.333333333333,66.666666666667)	1
##	[40,53.333333333333)	4
##	[26.666666666667,40)	2
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	1
##		
##	x	
##	y [40,53.333333333333)	
##	[186.666666666667,200]	1
##	[173.333333333333,186.666666666667)	7
##	[160,173.333333333333)	0
##	[146.666666666667,160)	1
##	[133.333333333333,146.666666666667)	3
##	[120,133.333333333333)	2
##	[106.666666666667,120)	2
##	[93.333333333333,106.666666666667)	4
##	[80,93.333333333333)	3
##	[66.666666666667,80)	3
##	[53.333333333333,66.666666666667)	0
##	[40,53.333333333333)	1
##	[26.666666666667,40)	1
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	2
##		
##	x	
##	y [53.333333333333,66.666666666667)	
##	[186.666666666667,200]	2
##	[173.333333333333,186.666666666667)	7
##	[160,173.333333333333)	0
##	[146.666666666667,160)	5
##	[133.333333333333,146.666666666667)	15
##	[120,133.333333333333)	2
##	[106.666666666667,120)	3

##	[93.333333333333,106.666666666667)	2
##	[80,93.333333333333)	3
##	[66.666666666667,80)	4
##	[53.333333333333,66.666666666667)	1
##	[40,53.333333333333)	0
##	[26.666666666667,40)	1
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	2
##		
##	x	
## y	[66.666666666667,80)	
##	[186.666666666667,200]	0
##	[173.333333333333,186.666666666667)	0
##	[160,173.333333333333)	8
##	[146.666666666667,160)	4
##	[133.333333333333,146.666666666667)	2
##	[120,133.333333333333)	0
##	[106.666666666667,120)	0
##	[93.333333333333,106.666666666667)	2
##	[80,93.333333333333)	3
##	[66.666666666667,80)	0
##	[53.333333333333,66.666666666667)	1
##	[40,53.333333333333)	2
##	[26.666666666667,40)	2
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	3
##		
##	x	
## y	[80,93.333333333333)	
##	[186.666666666667,200]	8
##	[173.333333333333,186.666666666667)	2
##	[160,173.333333333333)	5
##	[146.666666666667,160)	2
##	[133.333333333333,146.666666666667)	1
##	[120,133.333333333333)	5
##	[106.666666666667,120)	4
##	[93.333333333333,106.666666666667)	0
##	[80,93.333333333333)	3
##	[66.666666666667,80)	1
##	[53.333333333333,66.666666666667)	3
##	[40,53.333333333333)	0
##	[26.666666666667,40)	3
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	1
##		
##	x	
## y	[93.333333333333,106.666666666667)	
##	[186.666666666667,200]	3
##	[173.333333333333,186.666666666667)	3
##	[160,173.333333333333)	8
##	[146.666666666667,160)	8
##	[133.333333333333,146.666666666667)	20
##	[120,133.333333333333)	1
##	[106.666666666667,120)	2
##	[93.333333333333,106.666666666667)	3
##	[80,93.333333333333)	0
##	[66.666666666667,80)	5

##	[53.333333333333,66.666666666667)	3
##	[40,53.333333333333)	2
##	[26.666666666667,40)	2
##	[13.333333333333,26.666666666667)	0
##	[0,13.333333333333)	2
##	x	
## y	[106.666666666667,120)	
##	[186.666666666667,200]	1
##	[173.333333333333,186.666666666667)	1
##	[160,173.333333333333)	6
##	[146.666666666667,160)	2
##	[133.333333333333,146.666666666667)	4
##	[120,133.333333333333)	7
##	[106.666666666667,120)	1
##	[93.333333333333,106.666666666667)	2
##	[80,93.333333333333)	0
##	[66.666666666667,80)	3
##	[53.333333333333,66.666666666667)	5
##	[40,53.333333333333)	2
##	[26.666666666667,40)	0
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	2
##	x	
## y	[120,133.333333333333)	
##	[186.666666666667,200]	0
##	[173.333333333333,186.666666666667)	0
##	[160,173.333333333333)	0
##	[146.666666666667,160)	0
##	[133.333333333333,146.666666666667)	3
##	[120,133.333333333333)	6
##	[106.666666666667,120)	3
##	[93.333333333333,106.666666666667)	2
##	[80,93.333333333333)	3
##	[66.666666666667,80)	5
##	[53.333333333333,66.666666666667)	0
##	[40,53.333333333333)	1
##	[26.666666666667,40)	1
##	[13.333333333333,26.666666666667)	1
##	[0,13.333333333333)	0
##	x	
## y	[133.333333333333,146.666666666667)	
##	[186.666666666667,200]	3
##	[173.333333333333,186.666666666667)	0
##	[160,173.333333333333)	1
##	[146.666666666667,160)	4
##	[133.333333333333,146.666666666667)	4
##	[120,133.333333333333)	15
##	[106.666666666667,120)	4
##	[93.333333333333,106.666666666667)	5
##	[80,93.333333333333)	1
##	[66.666666666667,80)	4
##	[53.333333333333,66.666666666667)	0
##	[40,53.333333333333)	1
##	[26.666666666667,40)	2


```

## [13.333333333333,26.666666666667) 2
## [0,13.333333333333) 0
## x
## y [146.666666666667,160)
## [186.666666666667,200] 1
## [173.333333333333,186.666666666667) 2
## [160,173.333333333333) 0
## [146.666666666667,160) 6
## [133.333333333333,146.666666666667) 4
## [120,133.333333333333) 9
## [106.666666666667,120) 4
## [93.333333333333,106.666666666667) 5
## [80,93.333333333333) 4
## [66.666666666667,80) 3
## [53.333333333333,66.666666666667) 2
## [40,53.333333333333) 3
## [26.666666666667,40) 2
## [13.333333333333,26.666666666667) 0
## [0,13.333333333333) 0
## x
## y [160,173.333333333333)
## [186.666666666667,200] 1
## [173.333333333333,186.666666666667) 2
## [160,173.333333333333) 1
## [146.666666666667,160) 2
## [133.333333333333,146.666666666667) 9
## [120,133.333333333333) 8
## [106.666666666667,120) 2
## [93.333333333333,106.666666666667) 7
## [80,93.333333333333) 0
## [66.666666666667,80) 1
## [53.333333333333,66.666666666667) 1
## [40,53.333333333333) 2
## [26.666666666667,40) 4
## [13.333333333333,26.666666666667) 0
## [0,13.333333333333) 0
## x
## y [173.333333333333,186.666666666667)
## [186.666666666667,200] 3
## [173.333333333333,186.666666666667) 3
## [160,173.333333333333) 2
## [146.666666666667,160) 0
## [133.333333333333,146.666666666667) 1
## [120,133.333333333333) 3
## [106.666666666667,120) 2
## [93.333333333333,106.666666666667) 0
## [80,93.333333333333) 22
## [66.666666666667,80) 0
## [53.333333333333,66.666666666667) 1
## [40,53.333333333333) 2
## [26.666666666667,40) 0
## [13.333333333333,26.666666666667) 0
## [0,13.333333333333) 0
## x

```

```
## y [186.666666666667,200]
## [186.666666666667,200] 2
## [173.333333333333,186.666666666667) 12
## [160,173.333333333333) 0
## [146.666666666667,160) 2
## [133.333333333333,146.666666666667) 2
## [120,133.333333333333) 0
## [106.666666666667,120) 0
## [93.3333333333333,106.666666666667) 1
## [80,93.3333333333333) 3
## [66.6666666666667,80) 0
## [53.3333333333333,66.6666666666667) 1
## [40,53.3333333333333) 0
## [26.6666666666667,40) 0
## [13.3333333333333,26.6666666666667) 1
## [0,13.3333333333333) 2
```

```
quadrat.test(longleaf, 15, 15, method = "M", nsim = 1999)
```

```
##
## Conditional Monte Carlo test of CSR using quadrat counts
## Pearson X2 statistic
##
## data: longleaf
## X2 = 766.77, p-value = 0.001
## alternative hypothesis: two.sided
##
## Quadrats: 15 by 15 grid of tiles
```

Part 1a.

```
whiteoaks <- lansing[lansing$marks == "whiteoak"]
```

```
clarkevans(whiteoaks)
```

```
## naive Donnelly cdf
## 0.9522825 0.9334610 0.9495106
```

```
clarkevans.test(whiteoaks, correction = "Donnelly")
```

```
##
## Clark-Evans test
## Donnelly correction
## Monte Carlo test based on 999 simulations of CSR with fixed n
##
## data: whiteoaks
## R = 0.93346, p-value = 0.014
## alternative hypothesis: two-sided
```

The Clark Evans Test with Donnelly correction yields a p-value of 0.01 ($p < .05$). This indicates that the null hypothesis that the data is randomly distributed can be rejected.

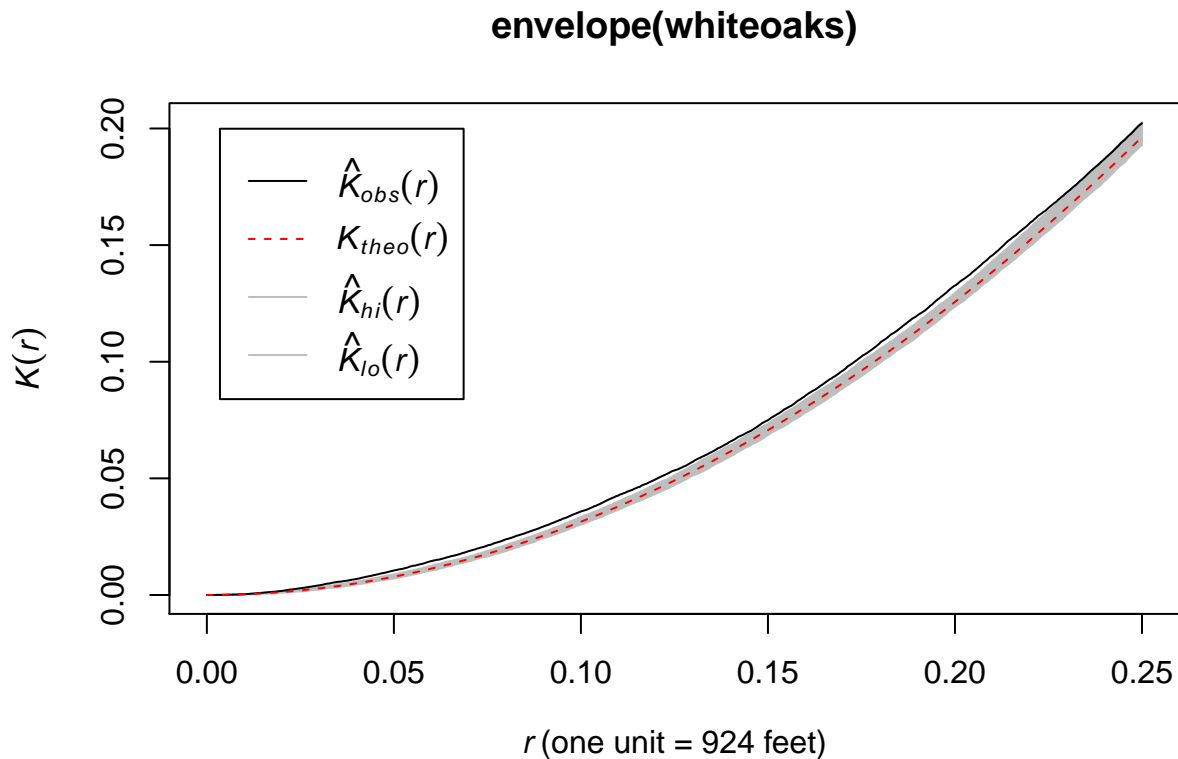
Part 1b.

Yes, the Clark and Evans Index is significantly different from 1.

Part 1c.

```
RipleyK <- Kest(whiteoaks)
plot(envelope(whiteoaks))
```

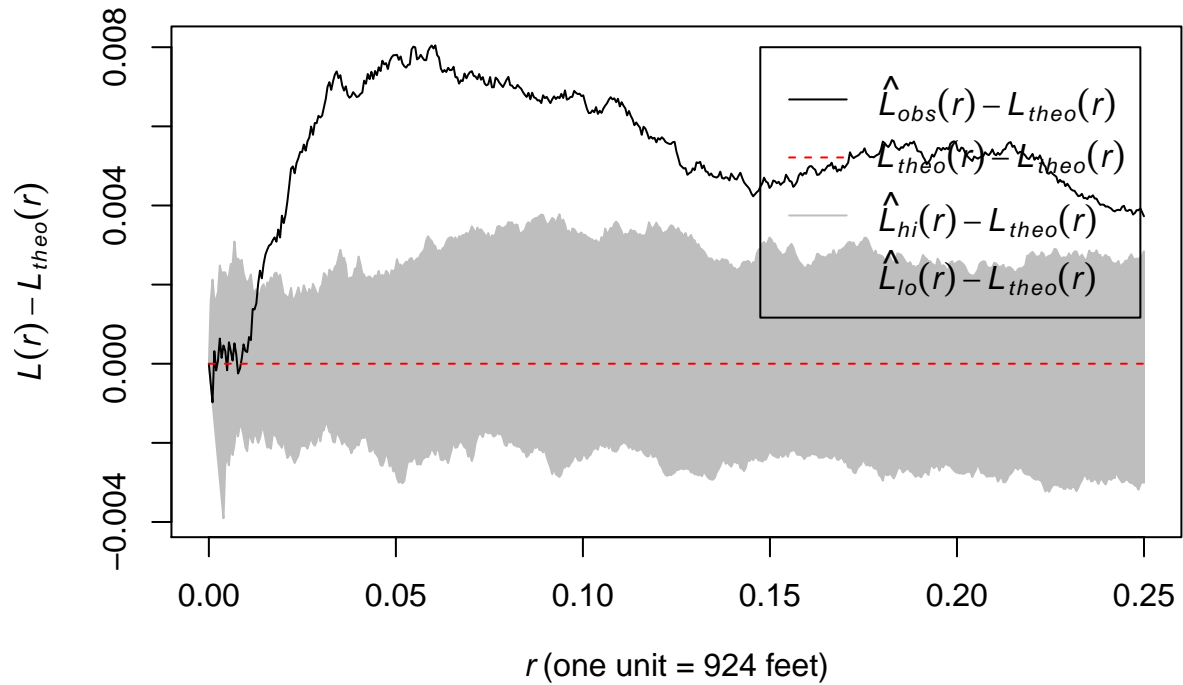
```
## Generating 99 simulations of CSR ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.
##
## Done.
```



```
RipleyL <- Lest(whiteoaks)
plot(envelope(whiteoaks, Lest, nsim = 99), .-theo~r)
```

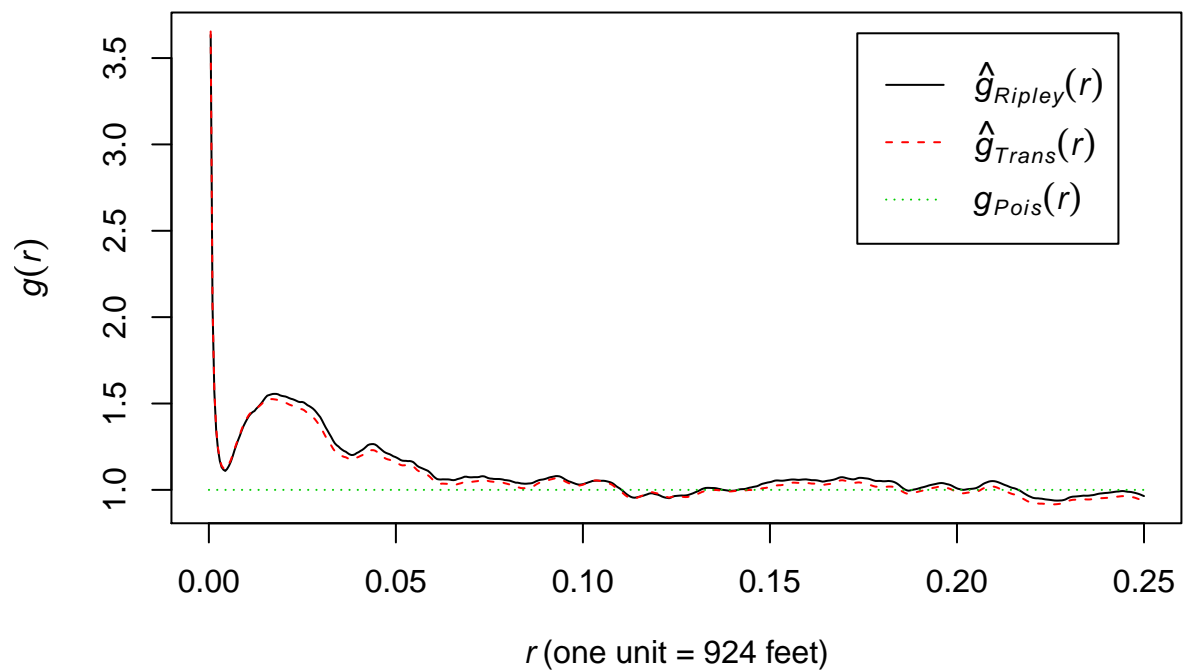
```
## Generating 99 simulations of CSR ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.
##
## Done.
```

envelope(whiteoaks, Lest, nsim = 99)



```
p <- pcf(whiteoaks)
plot(p)
```

p

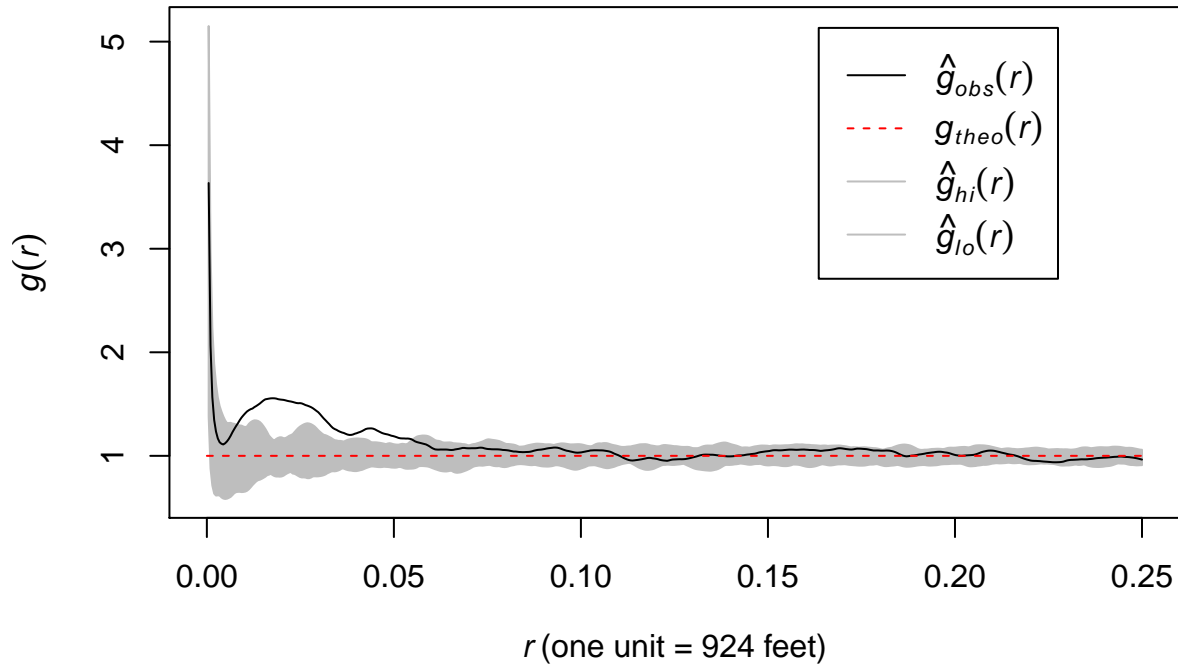


```
plot(envelope(whiteoaks, pcf, nsim = 99))
```

```
## Generating 99 simulations of CSR ...
```

```
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99.
##
## Done.
```

envelope(whiteoaks, pcf, nsim = 99)



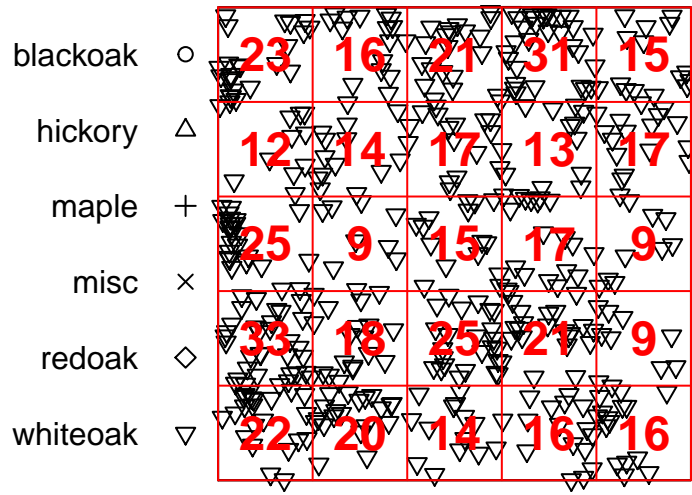
Part 1d.

```
QC <- quadratcount(whiteoaks)
QC
```

```
##           x
## y      [0,0.2) [0.2,0.4) [0.4,0.6) [0.6,0.8) [0.8,1]
## [0.8,1]      23      16      21      31      15
## [0.6,0.8)    12      14      17      13      17
## [0.4,0.6)    25       9      15      17       9
## [0.2,0.4)    33      18      25      21       9
## [0,0.2)     22      20      14      16      16
```

```
plot(whiteoaks)
plot(QC, add = TRUE, cex = 1.5, col = "red", font = 2)
```

whiteoaks



```
QC <- quadratcount(whiteoaks, 10, 10)
QC
```

```
##           x
## y      [0,0.1) [0.1,0.2) [0.2,0.3) [0.3,0.4) [0.4,0.5) [0.5,0.6)
## [0.9,1]         2         6         5         6         6         6
## [0.8,0.9)      13         2         1         4         7         2
## [0.7,0.8)       2         5         5         1         7         4
## [0.6,0.7)       1         4         6         2         2         4
## [0.5,0.6)      13         1         3         2         3         6
## [0.4,0.5)       8         3         2         2         3         3
## [0.3,0.4)       8         6         5         5         4         6
## [0.2,0.3)       9        10         6         2         5        10
## [0.1,0.2)      11         4        10         8         4         4
## [0,0.1)        2         5         1         1         2         4
##           x
## y      [0.6,0.7) [0.7,0.8) [0.8,0.9) [0.9,1]
## [0.9,1]        13         8         3         3
## [0.8,0.9)       6         4         4         5
## [0.7,0.8)       1         6         6         5
## [0.6,0.7)       4         2         4         2
## [0.5,0.6)       6         2         1         2
## [0.4,0.5)       4         5         4         2
## [0.3,0.4)       9         7         5         1
## [0.2,0.3)       2         3         1         2
## [0.1,0.2)       3         6        10         1
## [0,0.1)        1         6         4         1
```

```
quadrat.test(whiteoaks, 10, 10, method = "M", nsim = 1999)
```

```
##
## Conditional Monte Carlo test of CSR using quadrat counts
## Pearson X2 statistic
##
## data: whiteoaks
## X2 = 184.59, p-value = 0.001
```

```
## alternative hypothesis: two.sided
##
## Quadrats: 10 by 10 grid of tiles
```

Part 1e.

The p-value of the quadrat test is 0.001 ($p < .05$), indicating that we can reject the null hypothesis that the distribution of white oaks is random. The data indicates that the white oaks are not randomly distributed throughout the plot and may be either clumped or uniform. Visually, it looks as if the white oak distribution is clumped.