Homework 4

Stat 597a: Spatial Models

Claire Kelling

Due Nov 14, 2017

Problem 1a:

Load the data and create two ppp objects from it, one for affected and one for unaffected individuals. A key question is: what is the window? It is not available for this data. For now, take the window to be the same for each ppp object: use a rectangular region based on the range of x and y for both datasets. (Have a look at the help file for owin.)

I loaded the data and I created a plot of the data, seen below. I include whether the observations are from the affected or unaffected indivdiuals.

I also created these two ppp objects, please find the code in the appendix.

Full Data
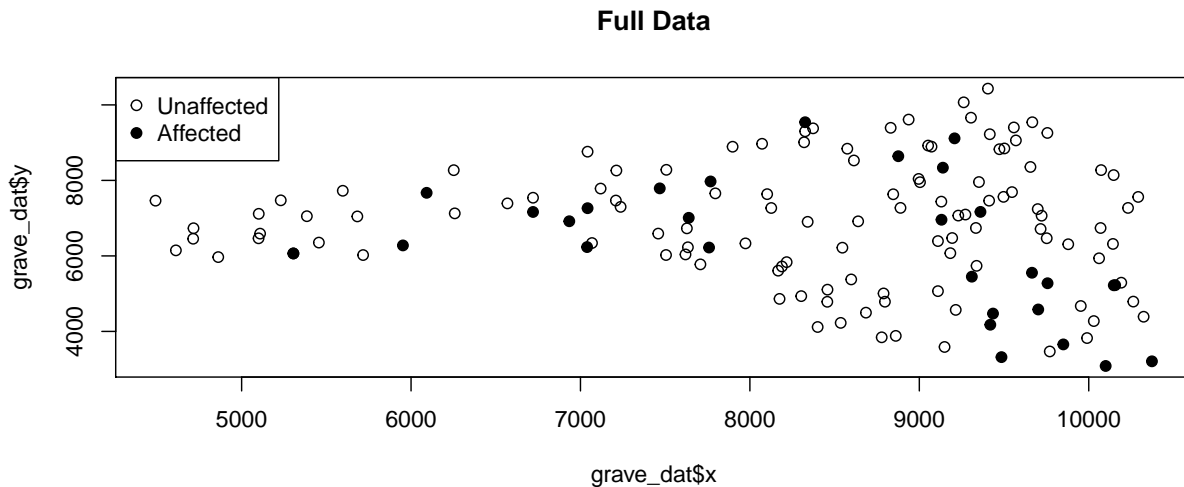
Problem 1b:

For each dataset separately, create Monte Carlo simulation envelopes for the F and G functions and plot them. Clearly label your plots and turn in a few sentences describing any choices you made in creating the envelopes. Is their evidence against CSR in this dataset? If so, what type of violation is suggested?

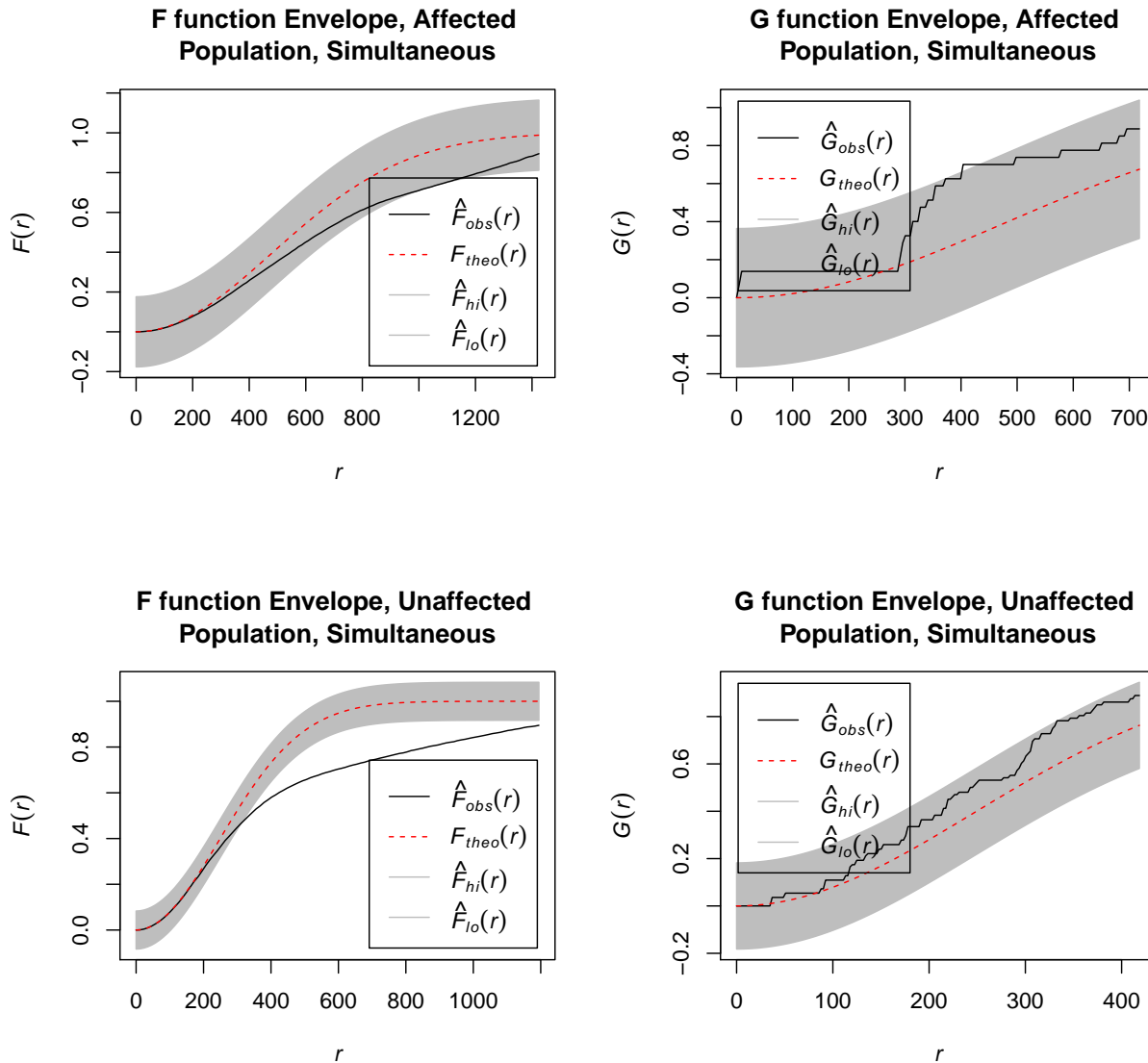I have included my plots below for the F and G functions for both datasets.

I have made a couple of different choices in creating the simulation envelopes. First, I will note that I include both the global = TRUE (simultaneous envelopes) and global = FALSE (pointwise envolepes) for both the affected and the unaffected individuals.
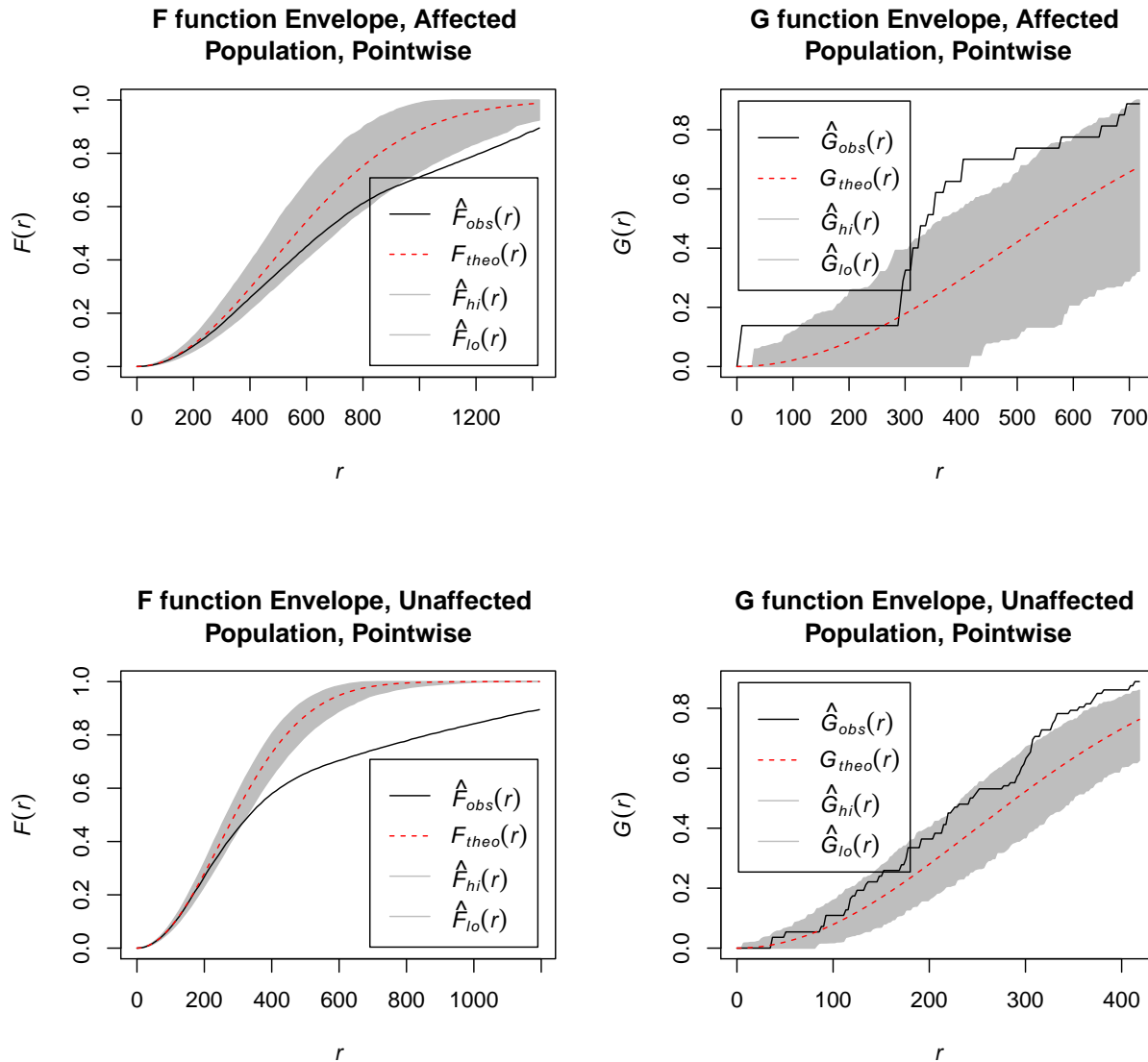
If I use the simultaneous envelopes, or set global = TRUE, I get a smoother function but a much more conservative estimate of CSR. I am somewhat concerned that these bands will always be too wide with the global =TRUE (compared to global = FALSE option) so that I will always fail to reject the null hypothesis

of complete spatial randomness. We see, for example, that the bands are quite wide around r=0 or a radius of 0, so that we shouldn't reject the null hypothesis of CSR if there are many points within a radius of 0, but this may support the case that there is no CSR. Therefore, I also include the pointwise envelopes, or global = FALSE.

I will also increase the number of simlations to 400 so that my results aren't impacted by a small number of simulations. I chose nrank to be different values depending on the global option, so that the 2.5% and the 97.5% quantiles of the simulations are displayed. We see from the help function that the significance level alpha for pointwise (global = FALSE) test is equal to 2*(nrank/(1+nsim)), so the significance level of my test is 5%. So for global = FALSE, I will have nrank = 10. For the simultaneous option (global = TRUE), the significance level alpha for the test is equal to nrank/(1+nsim). So for this option, I will have nrank = 20.

So, after creating my plots with these decisions, I don't see too much evidence against CSR in my dataset. For the most part, the observed curve lies within the simulation envelope for both that affected and the unaffected datasets, for both the F and G functions. However, for the unaffected dataset and the F curve, the observed curve is not very close to the mean simulated plot and lies outside the envelope on the right hand side of the plot.

F function Envelope, Affected Population, Pointwise

G function Envelope, Affected Population, Pointwise

F function Envelope, Unaffected Population, Pointwise

## G function Envelope, Unaffected Population, Pointwise

Problem 1c:

Now let's consider a more likely window: that these graves represent a complete excavation of the area in which they appear, and that area is irregularly shaped. Since we don't know what it is, we can choose a rough polygon outline to surround the points. You can create such an outline by plotting the locations and using the locator function; again, see help(owin) for the details of how to specify a polygon boundary. Create two new ppp objects with this new window.

Again, I created these two ppp objects, please find the code in the appendix.

Affected Individuals: New Window

**Unaffected Individuals: New Window**
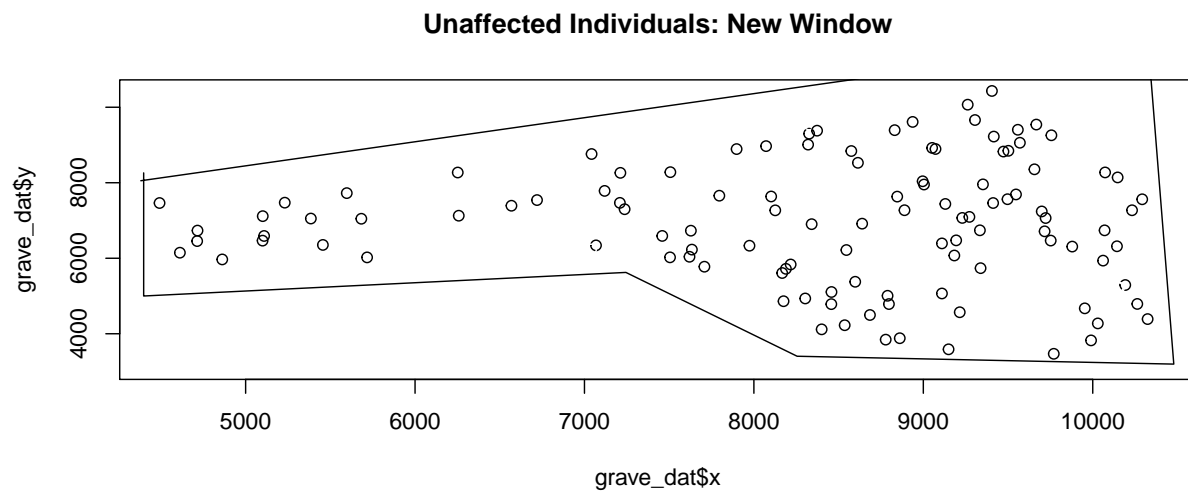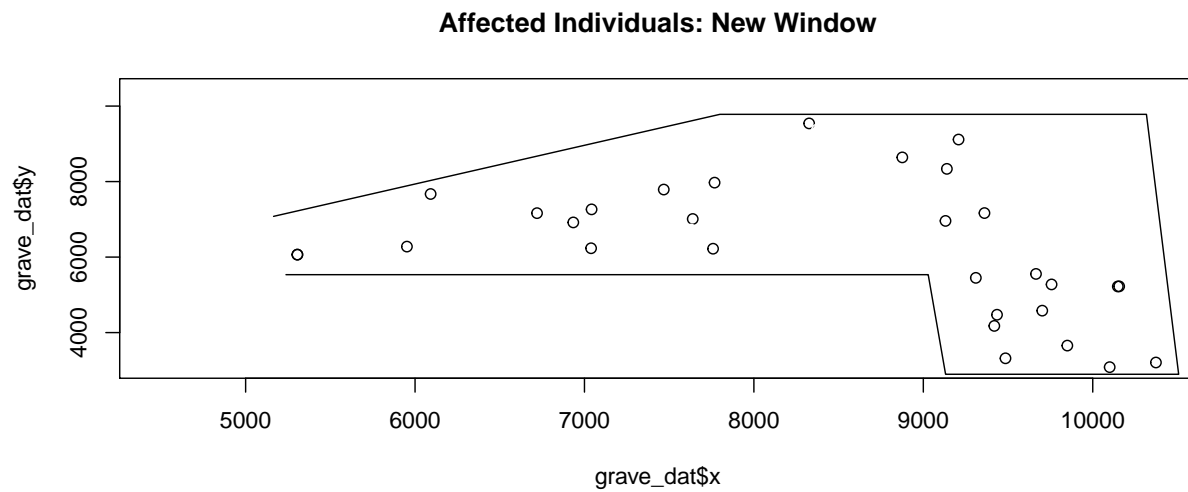
Problem 1d:

Repeat step (b) for the new datasets. What changes? Can you explain the reason for this, based on the form of the test statistics?

Again, I include the plots for both pointwise and simultaneous for the global option, just as in part b, with the same rank and nsim decisions.

Now that I have changed the window to polygons, the data is showing strong evidence for complete spatial randomness. The observed line is very close to the mean simulated line, especially in the F function. This is true for both the afffected and unaffected individuals.

I can see somewhat of the reason for this because of the test statistics. In part d, there is no evidence against complete spatial randomness, but in part b, the empiricaly F and G functions suggest repulsion among the points, which we can see from the F and G functions as we increase the radius r.
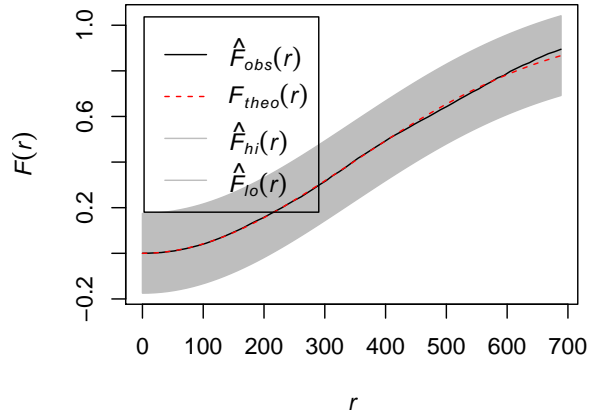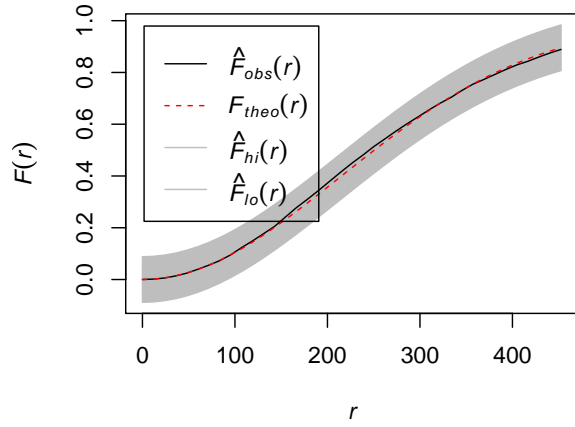
4

## F function Envelope, Affected Population, Simultaneous

$\hat{F}_{obs}(r)$
$F_{theo}(r)$
$\hat{F}_{hi}(r)$
$\hat{F}_{lo}(r)$

$F(r)$

r

G function Envelope, Affected Population, Simultaneous

$\hat{G}_{obs}(r)$
$G_{theo}(r)$
$\hat{G}_{hi}(r)$
$\hat{G}_{lo}(r)$

$G(r)$

r

F function Envelope, Unaffected Population, Simultaneous

$\hat{F}_{obs}(r)$
$F_{theo}(r)$
$\hat{F}_{hi}(r)$
$\hat{F}_{lo}(r)$

$F(r)$

r

G function Envelope, Unaffected Population, Simultaneous

$\hat{G}_{obs}(r)$
$G_{theo}(r)$
$\hat{G}_{hi}(r)$
$\hat{G}_{lo}(r)$

$G(r)$

r

F function Envelope, Affected Population, Pointwise

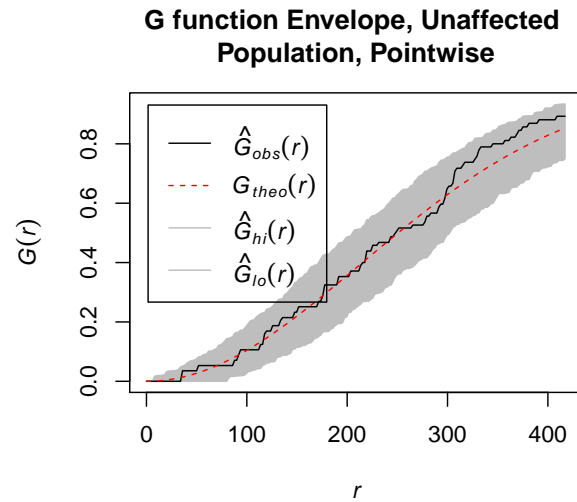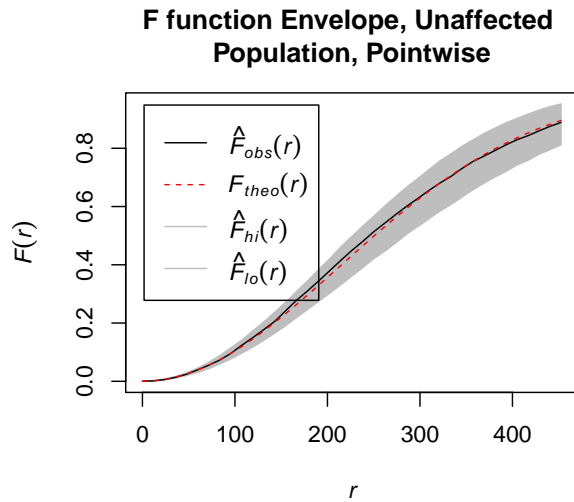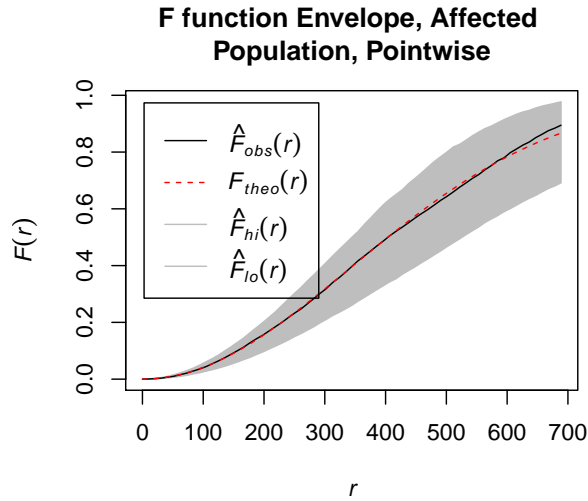**G function Envelope, Affected Population, Pointwise**
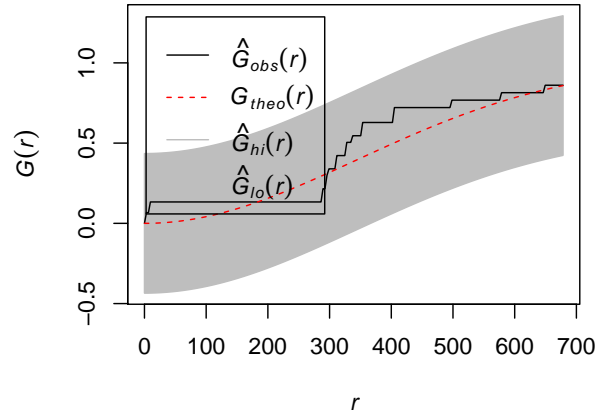
F function Envelope, Unaffected Population, Pointwise

**G function Envelope, Unaffected Population, Pointwise**
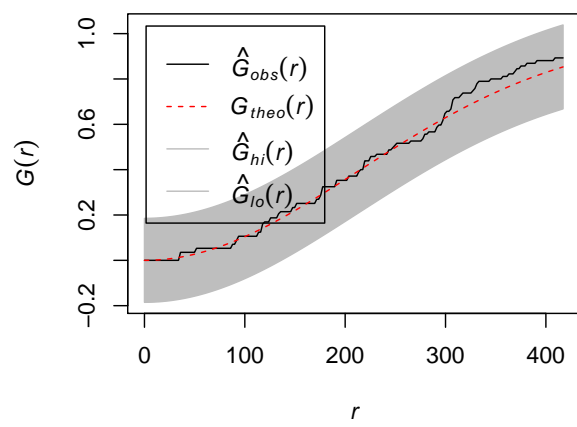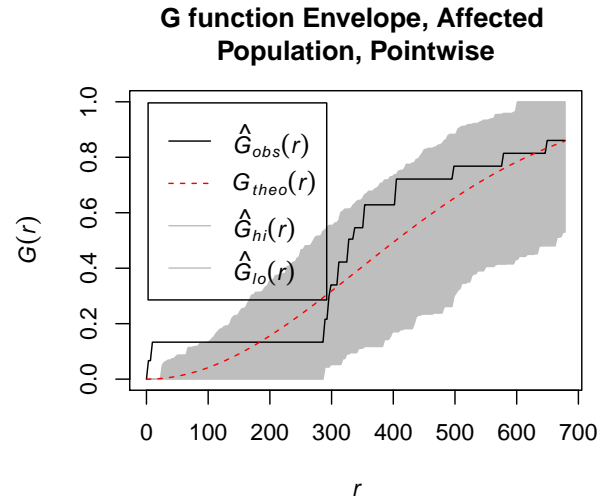
Problem 2:

Simulate four datasets on the unit square, from a homogeneous Poisson process with a rate λ of your choosing. For each one, fit a kernel estimate of the intensity function and plot it with the points overlaid. What do you conclude about interpreting the results of this estimator? In other words, what can go wrong if we fit a spatial point process model without checking for CSR first?

A homogenous Poisson Process is an example of a CSR point process model. If we do not assess for CSR first, we may conclude that there is not complete spatial randomness when we fit models. Even though they are estimators of the same intensity, they give vastly different results in terms of the estimator. In this case, the kernel density estimates suggest that the data comes from an inhomogenous poisson process, which is not true.
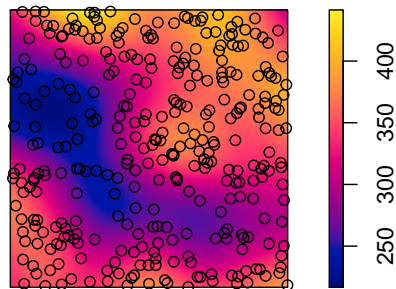
Kernel estimate with Uniform
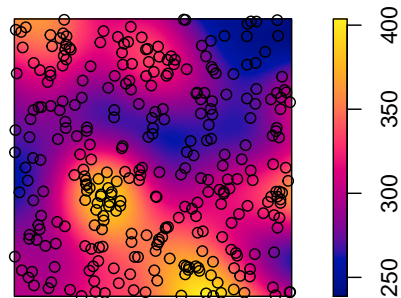Method on (0,1)x(0,1) when intensity = λ = 300
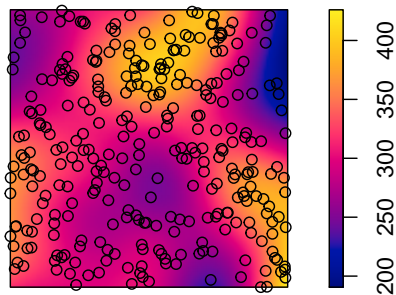
Kernel estimate with Uniform
Method on (0,1)x(0,1) when intensity = λ = 300
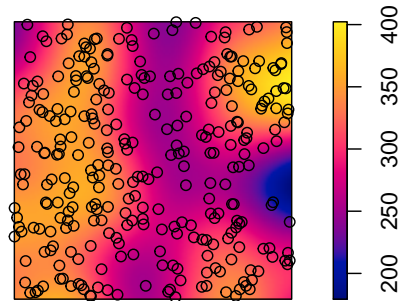
Kernel estimate with Uniform
Method on (0,1)x(0,1) when intensity = λ = 300

Kernel estimate with Uniform
Method on (0,1)x(0,1) when intensity = λ = 300

Problem 3:

Read through sections 15.2 and 15.3 of the notes by Adrian Baddeley about fitting point process models in R with the spatstat package (you can find it on Canvas). In 15.3, follow along with the R code example for the bei dataset. In particular, find the MLEs for inhomogeneous models with intensity functions:

$$\text{Model 1: } = exp\{\beta_0 + \beta_1 Z(x)\}$$

$$\text{Model 2: } = \beta Z(x)$$

Plot a kernel density estimate of $\lambda(x)$, ignoring covariates, as well as the fitted intensities under the two models. Put them all on the same color scale, and include a sentence or two comparing them. Note: the data for this problem are available from the spatstat package.

The first plot is a kernel density estimate, using density.ppp. The second plot is the plot that is an inhomogenous poisson process with the intensity that just depends on the spatial locations.

I notice that the kernel density estimate of $\lambda(x)$ is fairly different than the fitted intensities under the two models. The two models are relatively similar, just slightly different ranges. However, they include peaks at pretty much the same set of locations.

We also see that the inhomogenous poisson process with intensity proportional to the slope (third plot) fits the data better than one with an intensity that is loglinear function of the slope (second plot). This is is because it better captures teh variation of the intensity estimates across space, which we can see from the difference in range in the surface plot.
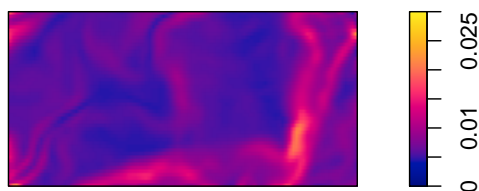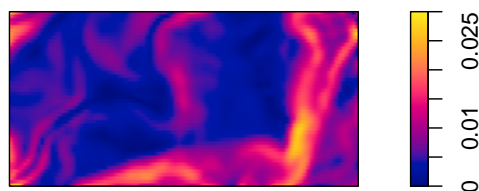
Kernel Density Estimate
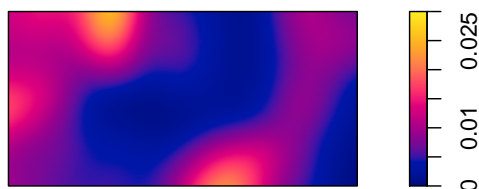
Basic Inhomogenous PP, No Covariates

Model 1

Model 2

Appendix: R Code

```
###
### Problem 1a
###
```

```r
#load the data
grave_dat <- read.table("C:/Users/ckell/OneDrive/Penn State/2017-2018/597/spatial_statistics_597/Homewo
#ignore the first column
grave_dat <- grave_dat[,-1]
#label the variables
colnames(grave_dat) <- c("indicator", "x", "y")

#I will include a plot of the data
plot(x=grave_dat$x, y = grave_dat$y, pch=ifelse(grave_dat$indicator ==1, 19, 1), col = "black", main =
legend("topleft", legend = c("Unaffected", "Affected"), pch = c(1,19), col = c("black", "black"))

#affected dataset
aff_dat <- grave_dat[which(grave_dat$indicator == 1),]
#unaffected dataset
unaff_dat <- grave_dat[which(grave_dat$indicator == 0),]

#what window to use? for now, take the window to be the same for each ppp object
#     the window will be a rectangular region based on the range of x and y for both datasets
window <- owin(xrange = range(grave_dat$x), yrange = range(grave_dat$y))

#create ppp object- affected individuals
aff_ppp <- as.ppp(aff_dat[,c(2:3)], W = window)

#create ppp object- unaffected individuals
unaff_ppp <- as.ppp(unaff_dat[,c(2:3)], W = window)

###
### Problem 1b
###

### SIMULTANEOUS, global = TRUE
#creating MC sim envelope for affected dataset
#     F function, simult
f_aff_sim <- envelope(aff_ppp, fun = Fest, global = TRUE, nrank = 20, nsim = 400)

#     G function, simult
g_aff_sim <- envelope(aff_ppp, fun = Gest, global = TRUE, nrank = 20, nsim = 400)


#creating MC sim envelope for unaffected dataset
#     F function, simult
f_unaff_sim <- envelope(unaff_ppp, fun = Fest, global = TRUE, nrank = 20, nsim = 400)

#     G function, simult
g_unaff_sim <- envelope(unaff_ppp, fun = Gest, global = TRUE, nrank = 20, nsim = 400)



### POINTWISE, global = FALSE
#creating MC sim envelope for affected dataset
#     F function, pointwise
f_aff_pnt <- envelope(aff_ppp, fun = Fest, global = FALSE, nrank = 10, nsim = 400)

#     G function, pointwise
```

```r
g_aff_pnt <- envelope(aff_ppp, fun = Gest, global = FALSE, nrank = 10, nsim = 400)


#creating MC sim envelope for unaffected dataset
#     F function, pointwise
f_unaff_pnt <- envelope(unaff_ppp, fun = Fest, global = FALSE, nrank = 10, nsim = 400)

#     G function, pointwise
g_unaff_pnt <- envelope(unaff_ppp, fun = Gest, global = FALSE, nrank = 10, nsim = 400)
```
```r
###
### Simultaneous
###
#creating MC sim plots for affected dataset
par(mfrow= c(1,2))
#     F function
plot(f_aff_sim, main = "F function Envelope, Affected \nPopulation, Simultaneous")
#     G function
plot(g_aff_sim, main = "G function Envelope, Affected \nPopulation, Simultaneous")

#creating MC sim plots for unaffected dataset
#     F function
plot(f_unaff_sim, main = "F function Envelope, Unaffected \nPopulation, Simultaneous")
#     G function
plot(g_unaff_sim, main = "G function Envelope, Unaffected \nPopulation, Simultaneous")



###
### Pointwise
###
#creating MC sim plots for affected dataset
#     F function
plot(f_aff_pnt, main = "F function Envelope, Affected \nPopulation, Pointwise")
#     G function
plot(g_aff_pnt, main = "G function Envelope, Affected \nPopulation, Pointwise")

#creating MC sim plots for unaffected dataset
#     F function
plot(f_unaff_pnt, main = "F function Envelope, Unaffected \nPopulation, Pointwise")
#     G function
plot(g_unaff_pnt, main = "G function Envelope, Unaffected \nPopulation, Pointwise")
```
```r
###
### Problem 1c
###

#create polygons - affected individuals
plot(aff_dat[,c(2,3)])
aff_poly <- locator()

#create polygons - affected individuals
plot(unaff_dat[,c(2,3)])
unaff_poly <- locator()
```

```r
save(aff_poly, unaff_poly, file = "C:/Users/ckell/OneDrive/Penn State/2017-2018/597/spatial_statistics_

###
### Problem 1d
###

### SIMULTANEOUS, global = TRUE
#creating MC sim envelope for affected dataset
#    F function, simult
f_aff_sim <- envelope(aff_ppp_poly, fun = Fest, global = TRUE, nrank = 20, nsim = 400)

#    G function, simult
g_aff_sim <- envelope(aff_ppp_poly, fun = Gest, global = TRUE, nrank = 20, nsim = 400)


#creating MC sim envelope for unaffected dataset
#    F function, simult
f_unaff_sim <- envelope(unaff_ppp_poly, fun = Fest, global = TRUE, nrank = 20, nsim = 400)

#    G function, simult
g_unaff_sim <- envelope(unaff_ppp_poly, fun = Gest, global = TRUE, nrank = 20, nsim = 400)



### POINTWISE, global = FALSE
#creating MC sim envelope for affected dataset
#    F function, pointwise
f_aff_pnt <- envelope(aff_ppp_poly, fun = Fest, global = FALSE, nrank = 10, nsim = 400)

#    G function, pointwise
g_aff_pnt <- envelope(aff_ppp_poly, fun = Gest, global = FALSE, nrank = 10, nsim = 400)


#creating MC sim envelope for unaffected dataset
#    F function, pointwise
f_unaff_pnt <- envelope(unaff_ppp_poly, fun = Fest, global = FALSE, nrank = 10, nsim = 400)

#    G function, pointwise
g_unaff_pnt <- envelope(unaff_ppp_poly, fun = Gest, global = FALSE, nrank = 10, nsim = 400)

par(mfrow= c(1,2))
###
### Simultaneous
###
#creating MC sim plots for affected dataset
#    F function
plot(f_aff_sim, main = "F function Envelope, Affected \nPopulation, Simultaneous")
#    G function
plot(g_aff_sim, main = "G function Envelope, Affected \nPopulation, Simultaneous")

#creating MC sim plots for unaffected dataset
#    F function
plot(f_unaff_sim, main = "F function Envelope, Unaffected \nPopulation, Simultaneous")
#    G function
```

```
plot(g_unaff_sim, main = "G function Envelope, Unaffected \nPopulation, Simultaneous")


###
### Pointwise
###
#creating MC sim plots for affected dataset
#    F function
plot(f_aff_pnt, main = "F function Envelope, Affected \nPopulation, Pointwise")
#    G function
plot(g_aff_pnt, main = "G function Envelope, Affected \nPopulation, Pointwise")

#creating MC sim plots for unaffected dataset
#    F function
plot(f_unaff_pnt, main = "F function Envelope, Unaffected \nPopulation, Pointwise")
#    G function
plot(g_unaff_pnt, main = "G function Envelope, Unaffected \nPopulation, Pointwise")

###
### Problem 2
###

#homogenous Poisson Process space and rate
x0=0;x1=1;y0=0;y1=1
lambda0<- 300
par(mfrow= c(1,2))

for(i in 1:4){
  #Simulating for number of events in the rectangle
  n.event=rpois(1,lambda0)

  #calculate x and y
  x.i=runif(n.event,min=x0,max=x1)
  y.i=runif(n.event,min=y0,max=y1)

  dat <- cbind(x.i, y.i)
  dat <- as.data.frame(dat)

  #Plot Homogenous Poisson Process
  #plot(dat,xlim=c(0,1), main=expression(paste("Uniform Method on D when intensity = ", lambda[0]," = 3

  #what window to use? for now, take the window to be the same for each ppp object
  #      the window will be a rectangular region of the unit square (0,1)x(0,1)
  window <- owin(xrange = c(0,1), yrange = c(0,1))


  #create ppp object- affected individuals
  dat_ppp <- as.ppp(dat, W = window)

  plot(density.ppp(dat_ppp), main = expression(paste("Kernel estimate with Uniform \nMethod on (0,1)x(0
  points(dat,xlim=c(0,1))
}
```

```
###
### Problem 3
###

#load the data from the spatstat package
data(bei)
grad <- bei.extra$grad

#plot the data
#plot(grad)

#model just based off of covariates
fit_basic <- ppm(bei, ~x+y)

#fit model 1
mod_1 <- ppm(bei, ~slope, covariates = list(slope = grad))
#mod_1

#fit model 2
mod_2 <- ppm(bei, ~offset(log(slope)), covariates = list(slope=grad))
#mod_2

par(mfrow=c(1,2))
# plot a kernel density estimate of lambda(x)
plot(density.ppp(bei), main = "Kernel Density Estimate", zlim = c(0.000,0.03))

#plot model with just x and y as predictors
plot(fit_basic, main = "Basic Inhomogenous PP, No Covariates",se= FALSE,ngrid= c(100,100), superimpose=F

# plot fitted intensity for model 1
plot(mod_1, se = FALSE,ngrid= c(100,100), superimpose = FALSE, main = "Model 1", zlim = c(0.000,0.03))

# plot fitted intensity for model 2
plot(mod_2, se = FALSE,ngrid= c(100,100), superimpose = FALSE, main = "Model 2", zlim = c(0.000,0.03))
```