Assignment 4: Spatial Point Patterns & Kriging

Lily Cheng

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(maps)
library(maptools)
```

```
## Loading required package: sp

## Warning: package 'sp' was built under R version 3.4.3

## Checking rgeos availability: TRUE
```

```r
library(sp)
library(spdep)
```

```
## Warning: package 'spdep' was built under R version 3.4.3

## Loading required package: Matrix

## Loading required package: spData

## Warning: package 'spData' was built under R version 3.4.3

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge')`
```

```r
library(gstat)
library(splancs)
```

```
##
## Spatial Point Pattern Analysis Code in S-Plus
##
##  Version 2 - Spatial and Space-Time analysis

##
## Attaching package: 'splancs'

## The following object is masked from 'package:dplyr':
##
##     tribble
```

```r
library(spatstat)
```

```
## Warning: package 'spatstat' was built under R version 3.4.3

## Loading required package: spatstat.data
```

```
## Warning: package 'spatstat.data' was built under R version 3.4.3

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## Loading required package: rpart

##
## spatstat 1.55-0       (nickname: 'Stunned Mullet')
## For an introduction to spatstat, type 'beginner'

## Warning in strptime(x, "%Y-%m-%d-%H-%M-%OS", tz = tz): unknown timezone
## 'zone/tz/2018c.1.0/zoneinfo/America/Los_Angeles'

##
## Attaching package: 'spatstat'

## The following object is masked from 'package:gstat':
##
##     idw
```

```
library(RColorBrewer) ## Visualization
library(classInt) ## Class intervals
library(spgwr) ## GWR (not sure we need this for now)
```

```
## Warning: package 'spgwr' was built under R version 3.4.2

## NOTE: This package does not constitute approval of GWR
## as a method of spatial analysis; see example(gwr)
```

```
library(lattice)
```

```
##
## Attaching package: 'lattice'

## The following object is masked from 'package:spatstat':
##
##     panel.histogram
```

```
library(rgdal) ## Geospatial data abstraction library
```

```
## Warning: package 'rgdal' was built under R version 3.4.3

## rgdal: version: 1.2-16, (SVN revision 701)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.1.3, released 2017/20/01
##  Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/3.4/Resources/library/rgdal/gdal
##  GDAL binary built with GEOS: FALSE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: /Library/Frameworks/R.framework/Versions/3.4/Resources/library/rgdal/pr
##  Linking to sp version: 1.2-5
```

```
my2k = readRDS('~/Desktop/Winter_2018/210B/Assignment4/h2kb.rds')
class(my2k)
```

```
## [1] "data.frame"
```

```r
summary(my2k)
```

```
##        Z1                  Z2             XCORD             YCORD
##  Min.   :   0.000   Min.   : 0.000   Min.   :-124.2   Min.   :32.56
##  1st Qu.:   5.524   1st Qu.: 2.000   1st Qu.:-121.9   1st Qu.:34.07
##  Median :  32.154   Median : 6.000   Median :-119.9   Median :36.59
##  Mean   :  68.084   Mean   : 8.414   Mean   :-120.0   Mean   :36.16
##  3rd Qu.:  78.969   3rd Qu.:12.000   3rd Qu.:-118.2   3rd Qu.:37.84
##  Max.   :1383.520   Max.   :95.000   Max.   :-114.4   Max.   :41.95
```

```r
data <- my2k
```

Create a matrix of coordinates from the point coordinates of the data file my2k (renamed data)

```r
sp_point <- cbind(data$XCORD, data$YCORD)
colnames(sp_point) <- c("LONG","LAT")
head(sp_point)
```

```
##           LONG      LAT
## [1,] -118.1920 34.06835
## [2,] -121.4452 38.63569
## [3,] -119.6622 36.34007
## [4,] -122.6362 38.22512
## [5,] -119.0181 35.32511
## [6,] -122.2539 37.85491
```

Part 1: Use the points I used for the examples in class (file h2kb.rds) and select two random samples of points from the California dataset of points (see page 7 of my notes). The first sample should be with 100 points (called herein SAMPLE1) and the second with 500 points (SAMPLE2). Compute the G function and F function for each sample and discuss the differences and commonalities between the two samples.

```r
proj <- CRS("+proj=utm +zone=10 +datum=WGS84")
summary(proj)
```

```
## Length  Class   Mode
##      1    CRS     S4
```

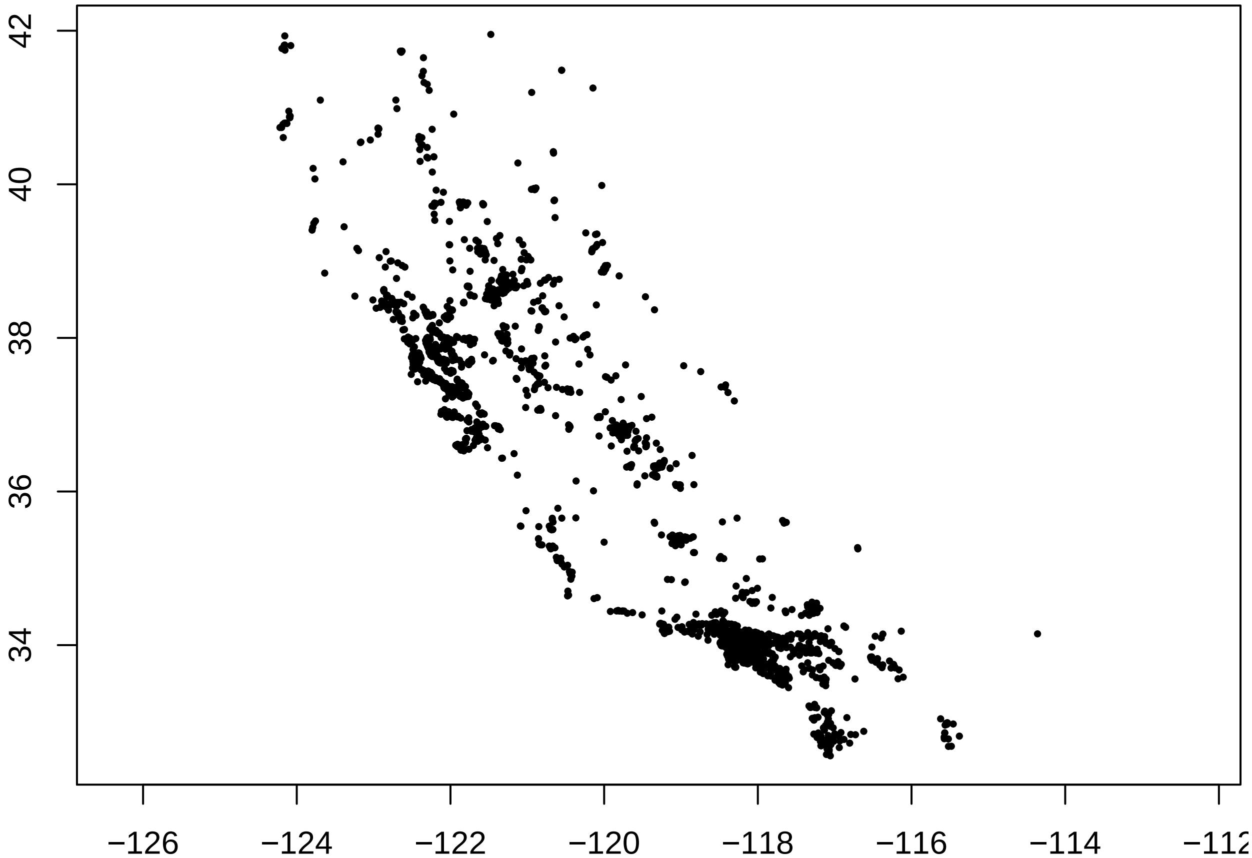
```r
data.sp <- SpatialPointsDataFrame(coords=sp_point,data,proj4string=proj)
summary(data.sp)
```

```
## Object of class SpatialPointsDataFrame
## Coordinates:
##             min        max
## LONG -124.21882 -114.36030
## LAT    32.55885   41.95175
## Is projected: TRUE
## proj4string :
## [+proj=utm +zone=10 +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0]
## Number of points: 2000
## Data attributes:
##        Z1                  Z2             XCORD             YCORD
##  Min.   :   0.000   Min.   : 0.000   Min.   :-124.2   Min.   :32.56
##  1st Qu.:   5.524   1st Qu.: 2.000   1st Qu.:-121.9   1st Qu.:34.07
##  Median :  32.154   Median : 6.000   Median :-119.9   Median :36.59
##  Mean   :  68.084   Mean   : 8.414   Mean   :-120.0   Mean   :36.16
##  3rd Qu.:  78.969   3rd Qu.:12.000   3rd Qu.:-118.2   3rd Qu.:37.84
##  Max.   :1383.520   Max.   :95.000   Max.   :-114.4   Max.   :41.95
```

```r
bbox(data.sp)
```

```
##              min        max
## LONG -124.21882 -114.36030
## LAT    32.55885   41.95175
```

```r
par(mar=c(2,2,0.2,0.2))
plot(data.sp,pch=16, cex=.5, axes=T)
```



SAMPLE1

```r
SAMPLE1 <- data[sample(100,replace=F),]
names(SAMPLE1)
```

```
## [1] "Z1"    "Z2"    "XCORD" "YCORD"
```

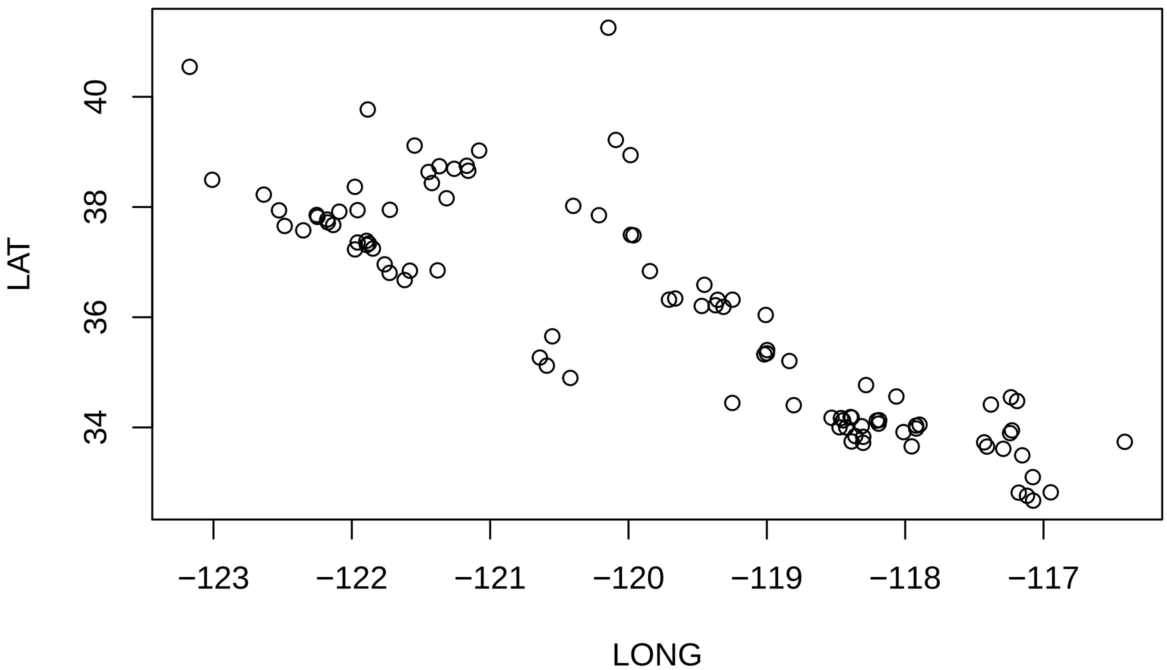
```r
head(SAMPLE1)
```

```
##               Z1 Z2     XCORD    YCORD
## 15835  43.312993  5 -116.9479 32.82246
## 27804  45.321728 14 -117.2354 34.54676
## 2480  168.133701 16 -119.6622 36.34007
## 29009 323.953437 37 -119.3138 36.18782
## 20364  45.567001  3 -118.3864 34.17982
## 21799   9.414831  4 -120.0915 39.21721
```

```r
dim(SAMPLE1)
```

```
## [1] 100   4
```

```
sp_point_SAMPLE1 <- matrix(NA, nrow=nrow(SAMPLE1),ncol=2)
sp_point_SAMPLE1[,1] <- SAMPLE1$XCORD
sp_point_SAMPLE1[,2] <- SAMPLE1$YCORD
colnames(sp_point_SAMPLE1) <- c("LONG","LAT")
plot(sp_point_SAMPLE1)
```




```
class(sp_point_SAMPLE1)
```

```
## [1] "matrix"
```

SAMPLE2

```
SAMPLE2 <- data[sample(500,replace=F),]
names(SAMPLE2)
```

```
## [1] "Z1"    "Z2"    "XCORD" "YCORD"
```

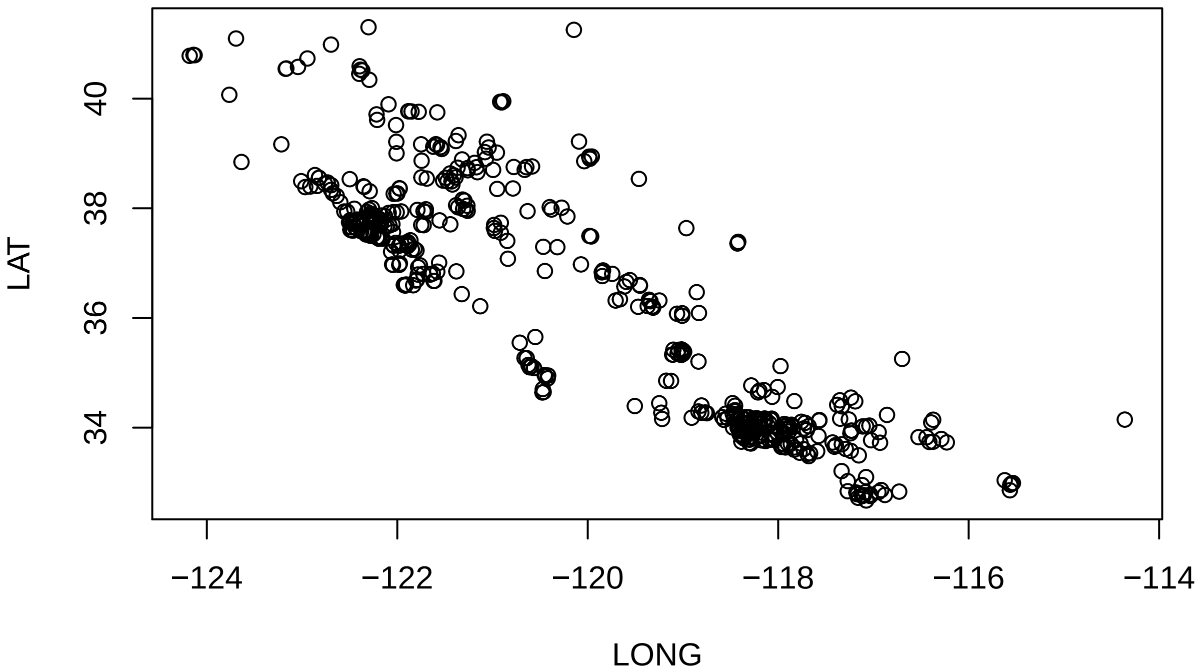
```
head(SAMPLE2)
```

```
##               Z1 Z2     XCORD    YCORD
## 10797 128.808554 19 -121.2812 37.97140
## 28662   0.000000  0 -118.7658 34.27635
## 1926    0.000000  0 -119.9853 38.94170
## 10463   8.063259  6 -118.4542 34.31756
## 12762   0.000000  2 -119.7424 36.80446
## 32615 140.710252 17 -121.3571 39.33306
```

```
dim(SAMPLE2)
```

```
## [1] 500   4
```

```
sp_point_SAMPLE2 <- matrix(NA, nrow=nrow(SAMPLE2),ncol=2)
sp_point_SAMPLE2[,1] <- SAMPLE2$XCORD
sp_point_SAMPLE2[,2] <- SAMPLE2$YCORD
colnames(sp_point_SAMPLE2) <- c("LONG","LAT")
plot(sp_point_SAMPLE2)
```




```
class(sp_point_SAMPLE2)
```

```
## [1] "matrix"
```

SAMPLE1 Create points that are uniformly distributed in space

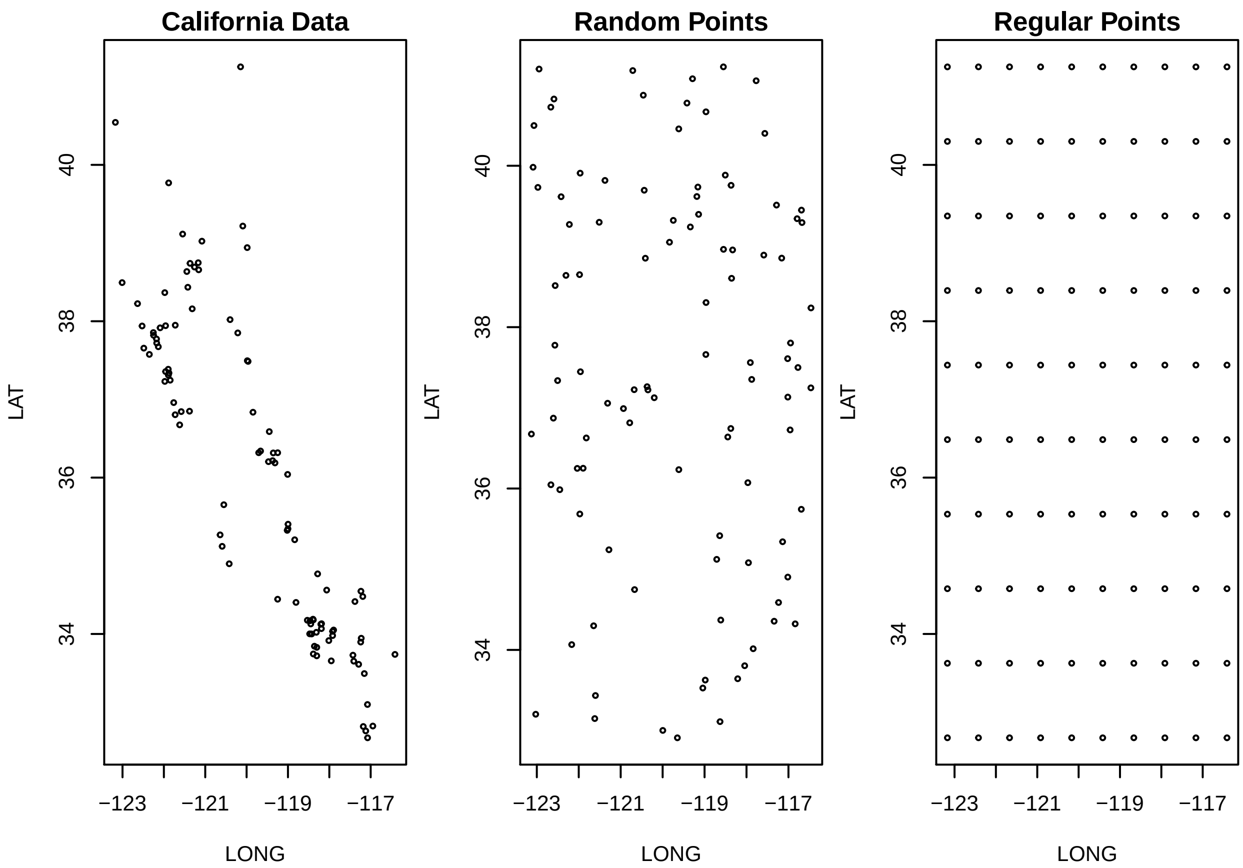
```
## Random points
u.x_SAMPLE1 <- runif(n=nrow(sp_point_SAMPLE1), min=bbox(sp_point_SAMPLE1)[1,1], max=bbox(sp_point_SAMPL
u.y_SAMPLE1 <- runif(n=nrow(sp_point_SAMPLE1), min=bbox(sp_point_SAMPLE1)[2,1], max=bbox(sp_point_SAMPL
```

SAMPLE1 Create points that are equispaced distributed in space

```
## Regular points
r.x_SAMPLE1 <- seq(from=min(sp_point_SAMPLE1[,1]),to=max(sp_point_SAMPLE1[,1]),
                   length=sqrt(nrow(sp_point_SAMPLE1)))
r.y_SAMPLE1 <- seq(from=min(sp_point_SAMPLE1[,2]),to=max(sp_point_SAMPLE1[,2]),
                   length=sqrt(nrow(sp_point_SAMPLE1)))
r.x_SAMPLE1 <- jitter(rep(r.x_SAMPLE1,length(r.x_SAMPLE1)),.001)
r.y_SAMPLE1 <- jitter(rep(r.y_SAMPLE1,each=length(r.y_SAMPLE1)),.001)
```

SAMPLE1 Plot the points to compare visually

```
## Plot the points
par(mfrow=c(1,3),mar=c(4,4,1.5,0.5))
plot(x=sp_point_SAMPLE1[,1],y=sp_point_SAMPLE1[,2],main="California Data", xlab="LONG",ylab="
LAT",cex=.5)
plot(x=u.x_SAMPLE1,y=u.y_SAMPLE1,main="Random Points", xlab="LONG",ylab="LAT",cex=.5)
plot(x=r.x_SAMPLE1,y=r.y_SAMPLE1,main="Regular Points", xlab="LONG",ylab="LAT",cex=.5)
```



SAMPLE2 Create points that are uniformly distributed in space

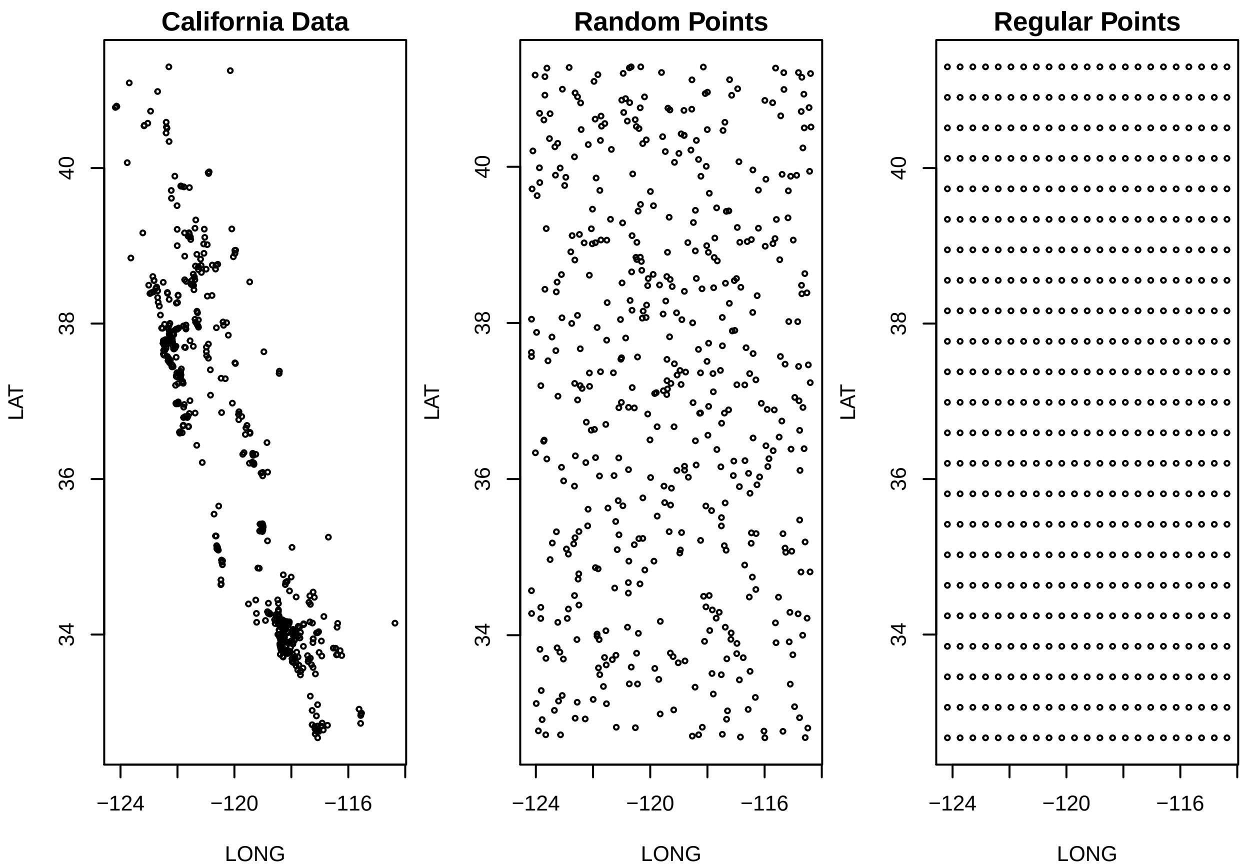
```
## Random points
u.x_SAMPLE2 <- runif(n=nrow(sp_point_SAMPLE2), min=bbox(sp_point_SAMPLE2)[1,1], max=bbox(sp_point_SAMPL
u.y_SAMPLE2 <- runif(n=nrow(sp_point_SAMPLE2), min=bbox(sp_point_SAMPLE2)[2,1], max=bbox(sp_point_SAMPL
```

SAMPLE2 Create points that are equispaced distributed in space

```
## Regular points
r.x_SAMPLE2 <- seq(from=min(sp_point_SAMPLE2[,1]),to=max(sp_point_SAMPLE2[,1]),
                   length=sqrt(nrow(sp_point_SAMPLE2)))
r.y_SAMPLE2 <- seq(from=min(sp_point_SAMPLE2[,2]),to=max(sp_point_SAMPLE2[,2]),
                   length=sqrt(nrow(sp_point_SAMPLE2)))
r.x_SAMPLE2 <- jitter(rep(r.x_SAMPLE2,length(r.x_SAMPLE2)),.001)
r.y_SAMPLE2 <- jitter(rep(r.y_SAMPLE2,each=length(r.y_SAMPLE2)),.001)
```

SAMPLE2 Plot the points to compare visually

```
## Plot the points
par(mfrow=c(1,3),mar=c(4,4,1.5,0.5))
plot(x=sp_point_SAMPLE2[,1],y=sp_point_SAMPLE2[,2],main="California Data", xlab="LONG",ylab="
LAT",cex=.5)
plot(x=u.x_SAMPLE2,y=u.y_SAMPLE2,main="Random Points", xlab="LONG",ylab="LAT",cex=.5)
plot(x=r.x_SAMPLE2,y=r.y_SAMPLE2,main="Regular Points", xlab="LONG",ylab="LAT",cex=.5)
```



Create function of conversion between kilometers and degrees. Function km2d (km to degrees)

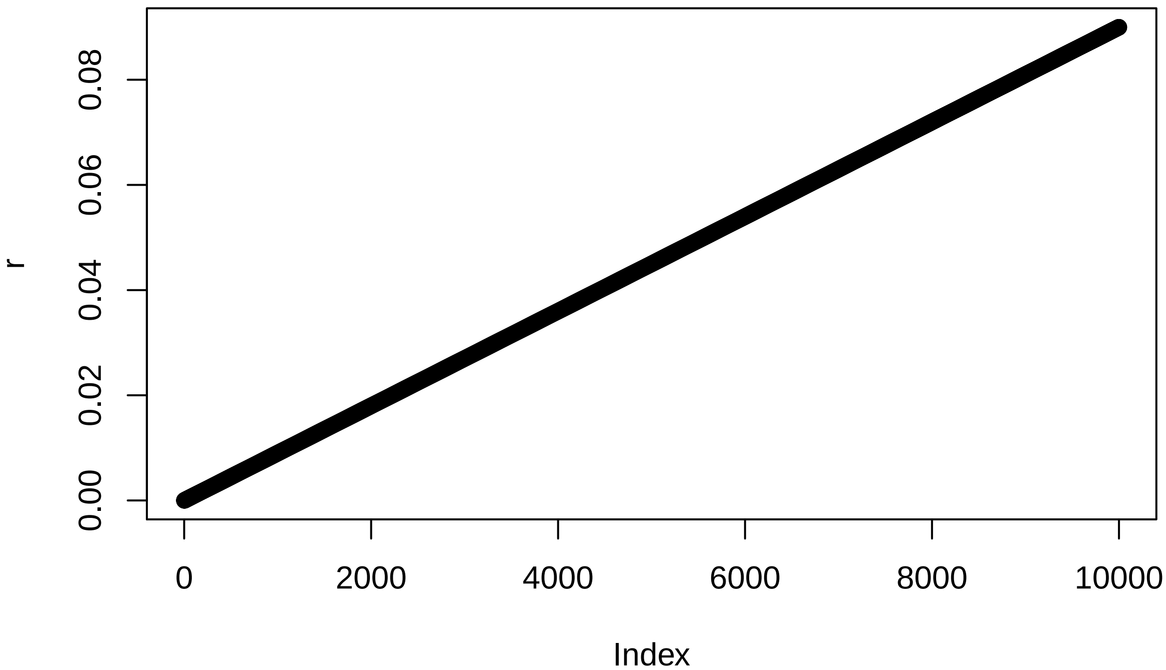
```
km2d <- function(km){
out <- (km/1.852)/60
return(out)
}
```

Function d2km (degrees to km)

```
d2km <- function(d){
out <- d*60*1.852
return(out)
}
```

Create a sequence of distances to compute G

```
r <- seq(0,km2d(10),length.out=10000)
plot(r)
```

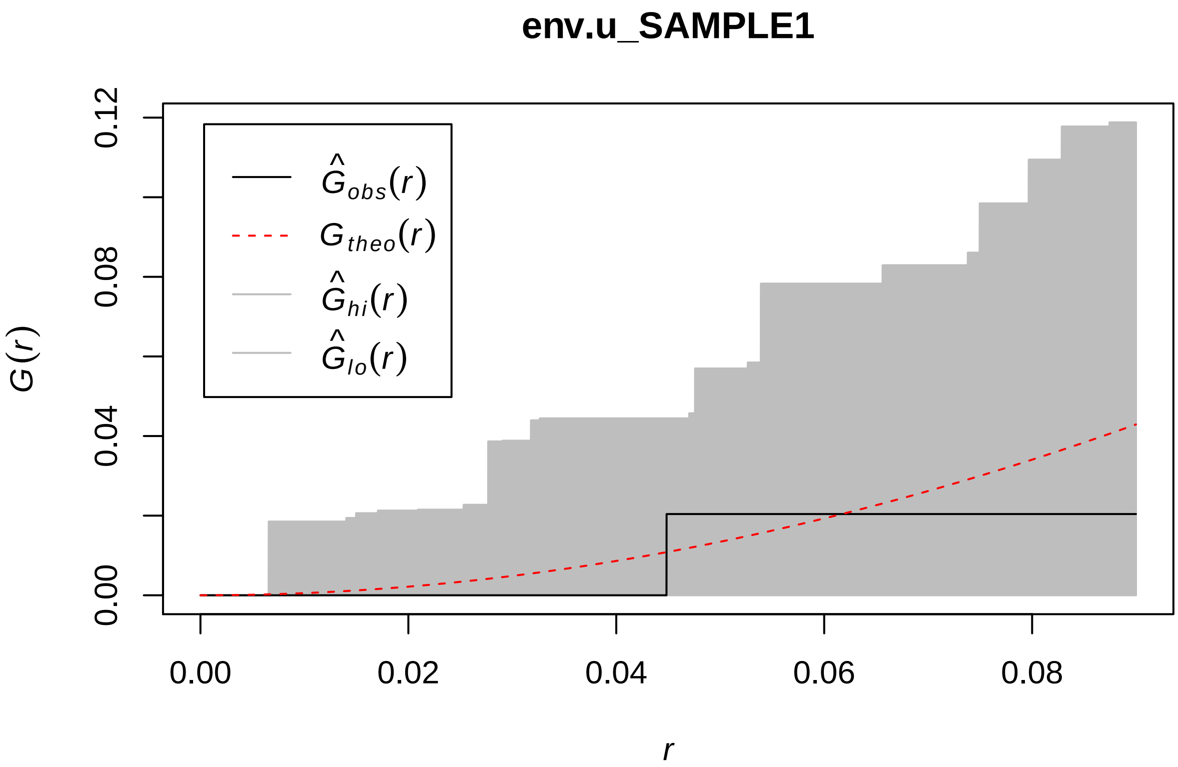


G-Test: California points vs Uniformly distributed points

```
env.u_SAMPLE1 <- envelope(ppp(x=u.x_SAMPLE1,y=u.y_SAMPLE1,window=owin(bbox(sp_point_SAMPLE1)[1,],bbox(sp
```

```
## 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
## 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, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

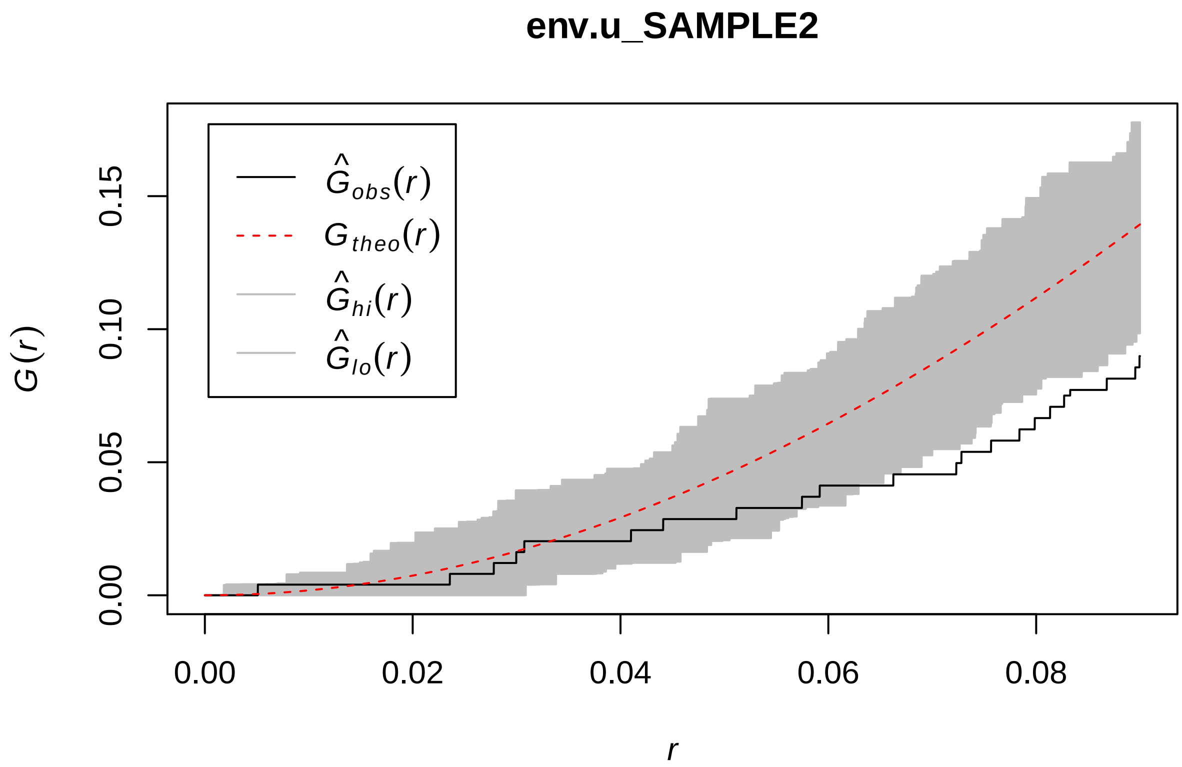
```
plot(env.u_SAMPLE1)
```




```
env.u_SAMPLE2 <- envelope(ppp(x=u.x_SAMPLE2,y=u.y_SAMPLE2,window=owin(bbox(sp_point_SAMPLE2)[1,],bbox(sp
```

```
## 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
## 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, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

```
plot(env.u_SAMPLE2)
```




```
## G-Test: California points
r <- seq(0,km2d(10),length.out=10000)
env_SAMPLE1 <- envelope(ppp(x=sp_point_SAMPLE1[,1],y=sp_point_SAMPLE1[,2],window=owin(bbox(sp_point_SAMI
```

```
## 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, 2
## 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, 
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

```
env_SAMPLE2 <- envelope(ppp(x=sp_point_SAMPLE2[,1],y=sp_point_SAMPLE2[,2],window=owin(bbox(sp_point_SAMI
```

```
## 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, 2
## 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, 
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

```
summary(env_SAMPLE1)
```

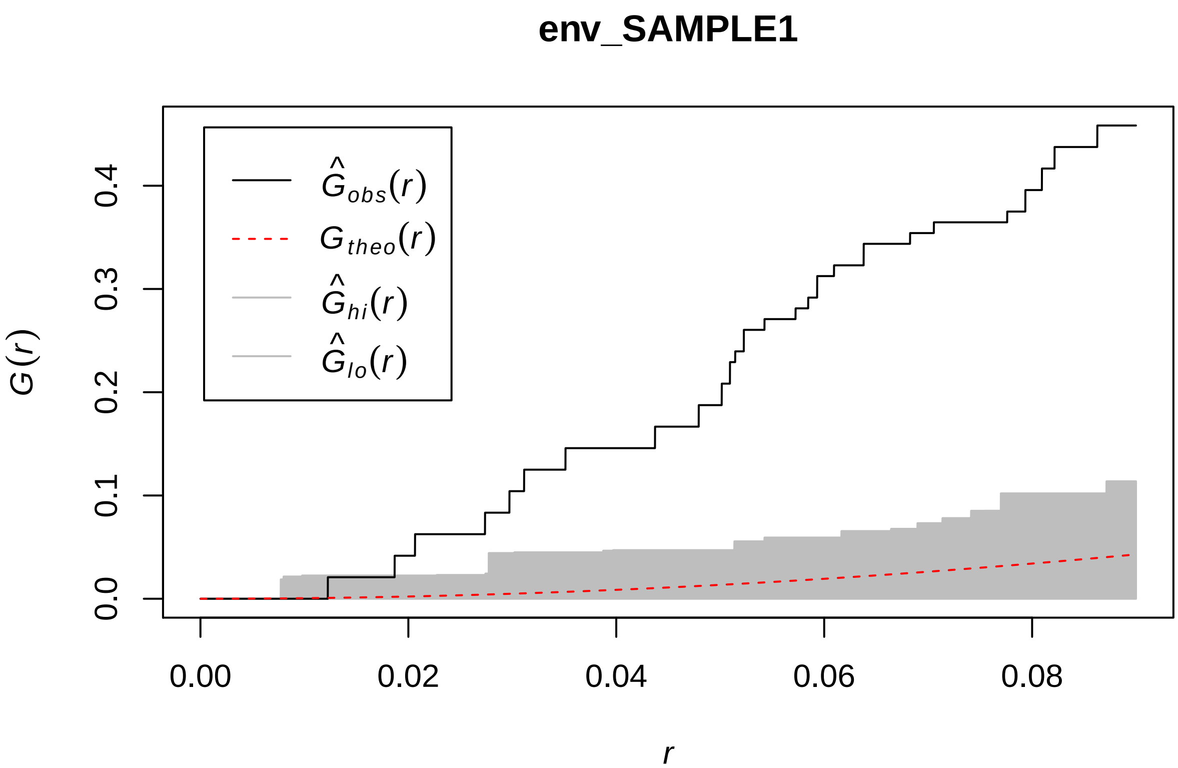
```
## Pointwise critical envelopes for G(r)
## and observed value for 'ppp(x = sp_point_SAMPLE1[, 1], y =
## sp_point_SAMPLE1[, 2], window = owin(bbox(sp_point_SAMPLE1)[1, '
## Obtained from 99 simulations of CSR
## Alternative: two.sided
## Upper envelope: pointwise 2nd largest of simulated curves
## Lower envelope: pointwise 2nd smallest of simulated curves
## Significance level of Monte Carlo test: 4/100 = 0.04
## Data: ppp(x = sp_point_SAMPLE1[, 1], y = sp_point_SAMPLE1[, 2], window =
```

```
## owin(bbox(sp_point_SAMPLE1)[1,
```

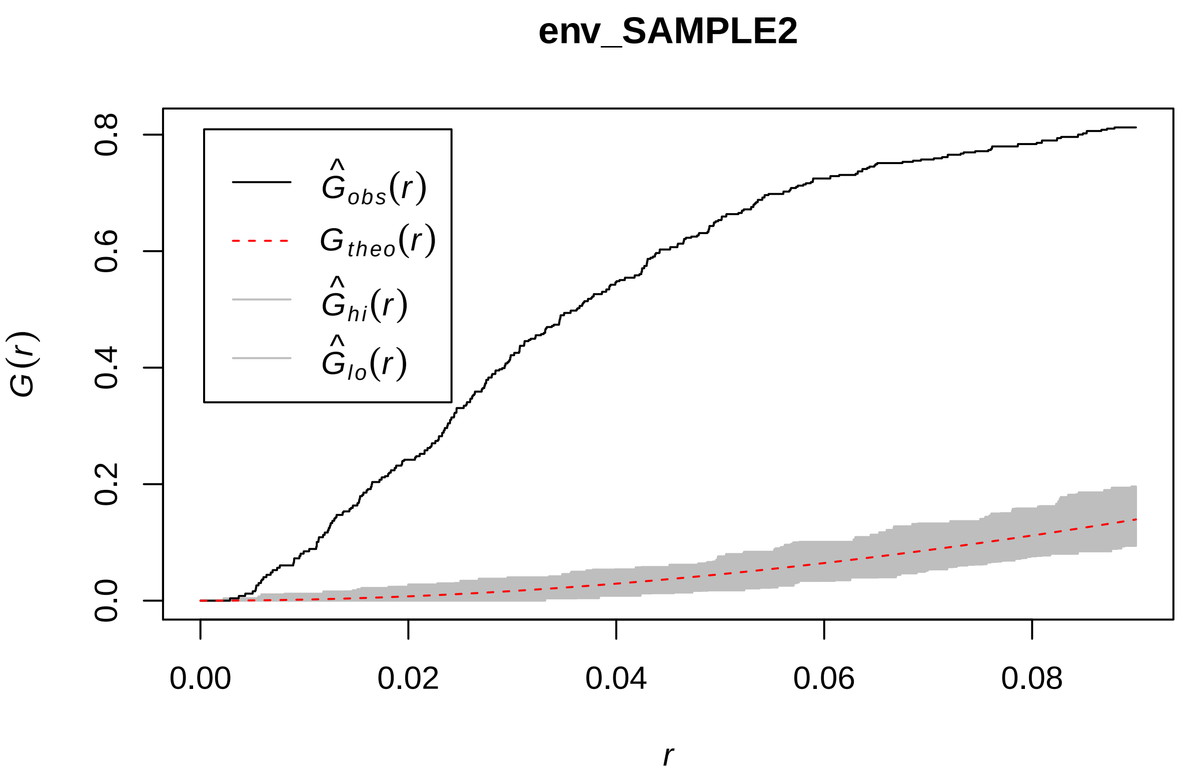
```r
summary(env_SAMPLE2)
```

```
## Pointwise critical envelopes for G(r)
## and observed value for 'ppp(x = sp_point_SAMPLE2[, 1], y =
## sp_point_SAMPLE2[, 2], window = owin(bbox(sp_point_SAMPLE2)[1, '
## Obtained from 99 simulations of CSR
## Alternative: two.sided
## Upper envelope: pointwise 2nd largest of simulated curves
## Lower envelope: pointwise 2nd smallest of simulated curves
## Significance level of Monte Carlo test: 4/100 = 0.04
## Data: ppp(x = sp_point_SAMPLE2[, 1], y = sp_point_SAMPLE2[, 2], window =
## owin(bbox(sp_point_SAMPLE2)[1,
```

```r
plot(env_SAMPLE1)
```




```r
plot(env_SAMPLE2)
```



Commonalities: in the above plots, (1) G values from both data SAMPLE1 and data SAMPLE2 are above the randomization envelope. Therefore, G Functions of both SAMPLE1 and SAMPLE2 show there is clustering in the data. (2) simulated curve of pointwise critical envelope is the same for both SAMPLE1 and SAMPLE2.

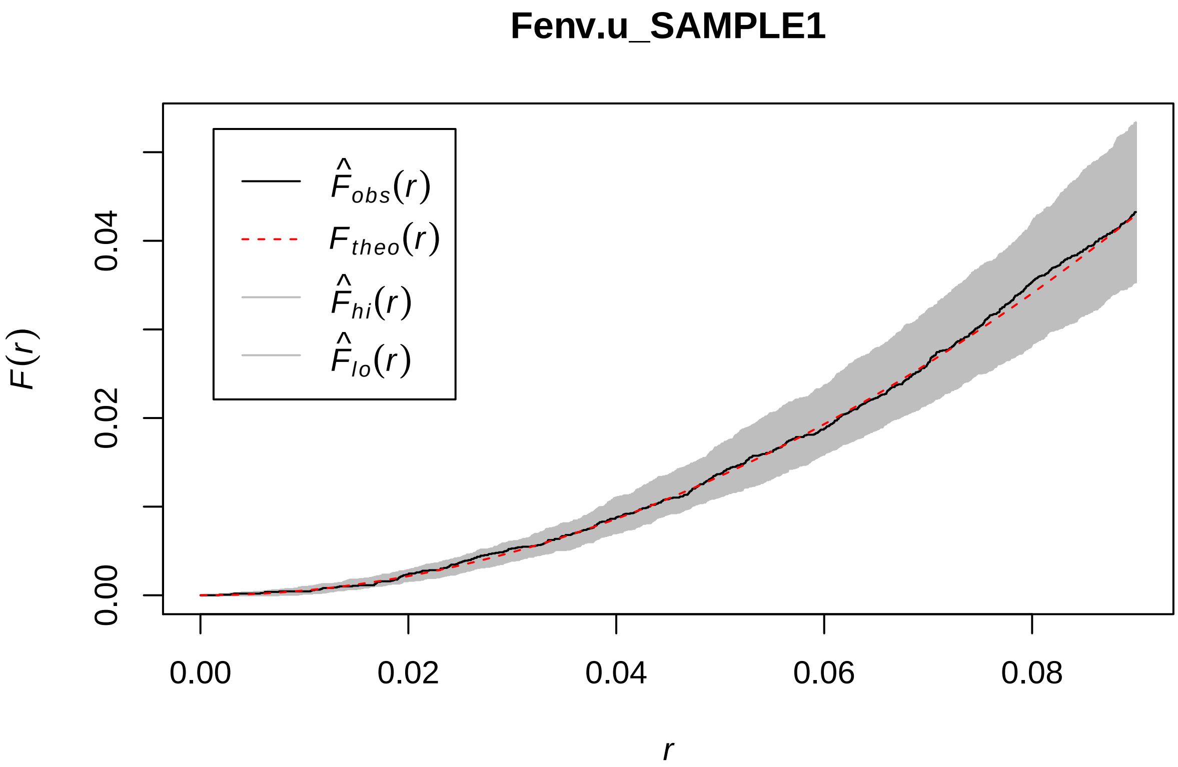
Differences: in the above plots, (1) the curve for G values of SAMPLE1 is less smooth and less curvy than SAMPLE2. (2) The lower envelope for SAMPLE1 is more constant and lower than the lower envelope of SAMPLE2. The upper envelope for SAMPLE1 is more choppy and higher than SAMPLE2. Thus, the range between upper envelope and lower envelope of SAMPLE1 is bigger than SAMPLE2 which means there is higher variablility in randomazation envelope of SAMPLE1.

F function

```
Fenv.u_SAMPLE1 <- envelope(ppp(x=u.x_SAMPLE1,y=u.y_SAMPLE1,window=owin(bbox(sp_point_SAMPLE1)[1,],bbox(s
```

```
## 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
## 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, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

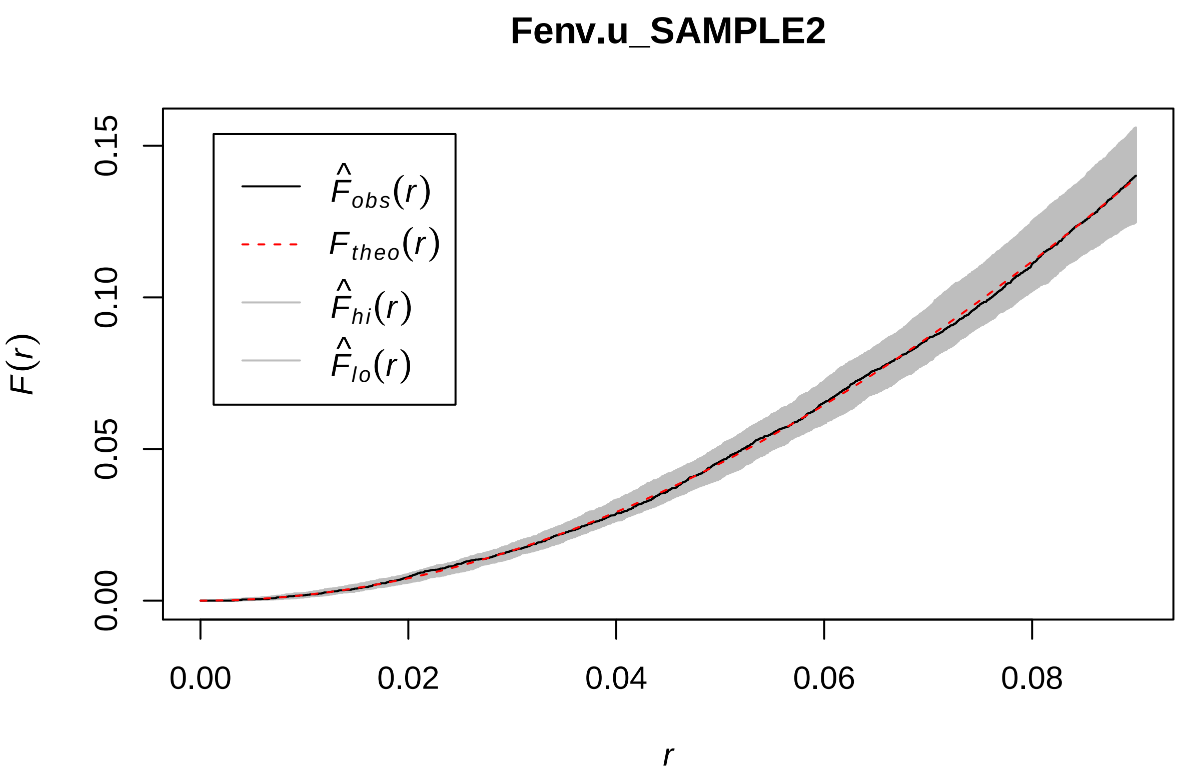
```
plot(Fenv.u_SAMPLE1)
```




```r
Fenv.u_SAMPLE2 <- envelope(ppp(x=u.x_SAMPLE2,y=u.y_SAMPLE2,window=owin(bbox(sp_point_SAMPLE2)[1,],bbox(s
```

```
## 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
## 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, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

```r
plot(Fenv.u_SAMPLE2)
```




```
## F-Test: California points
r <- seq(0,km2d(10),length.out=10000)
Fenv_SAMPLE1 <- envelope(ppp(x=sp_point_SAMPLE1[,1],y=sp_point_SAMPLE1[,2],
                             window=owin(bbox(sp_point_SAMPLE1)[1,],bbox(sp_point_SAMPLE1)[2,])),
                         fun=Fest, r=r, nsim=99, nrank=2)
```

```
## 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
## 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, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

```
summary(Fenv_SAMPLE1)
```

```
## Pointwise critical envelopes for F(r)
## and observed value for 'ppp(x = sp_point_SAMPLE1[, 1], y =
## sp_point_SAMPLE1[, 2], window = owin(bbox(sp_point_SAMPLE1)[1, '
## Obtained from 99 simulations of CSR
## Alternative: two.sided
## Upper envelope: pointwise 2nd largest of simulated curves
## Lower envelope: pointwise 2nd smallest of simulated curves
## Significance level of Monte Carlo test: 4/100 = 0.04
## Data: ppp(x = sp_point_SAMPLE1[, 1], y = sp_point_SAMPLE1[, 2], window =
## owin(bbox(sp_point_SAMPLE1)[1,
```

```
Fenv_SAMPLE2 <- envelope(ppp(x=sp_point_SAMPLE2[,1],y=sp_point_SAMPLE2[,2],
                             window=owin(bbox(sp_point_SAMPLE2)[1,],bbox(sp_point_SAMPLE2)[2,])),
                         fun=Fest, r=r, nsim=99, nrank=2)
```

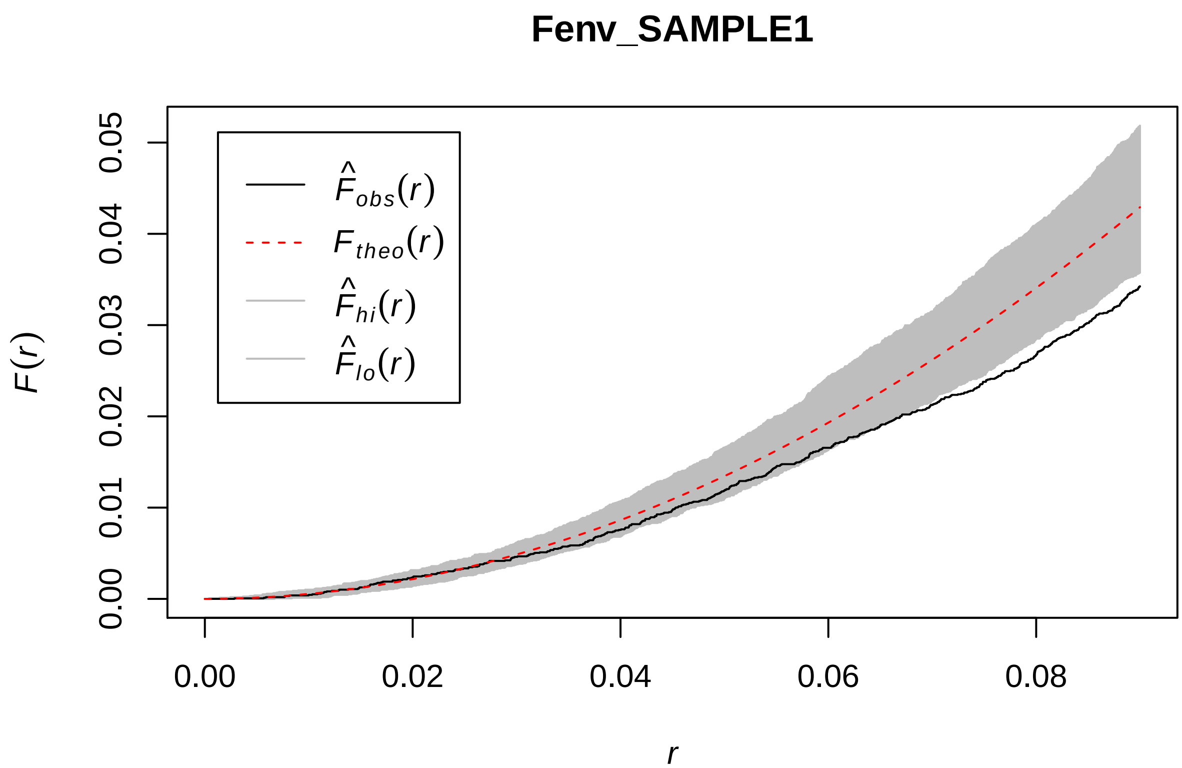
```
## Generating 99 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
```

```
## 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, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

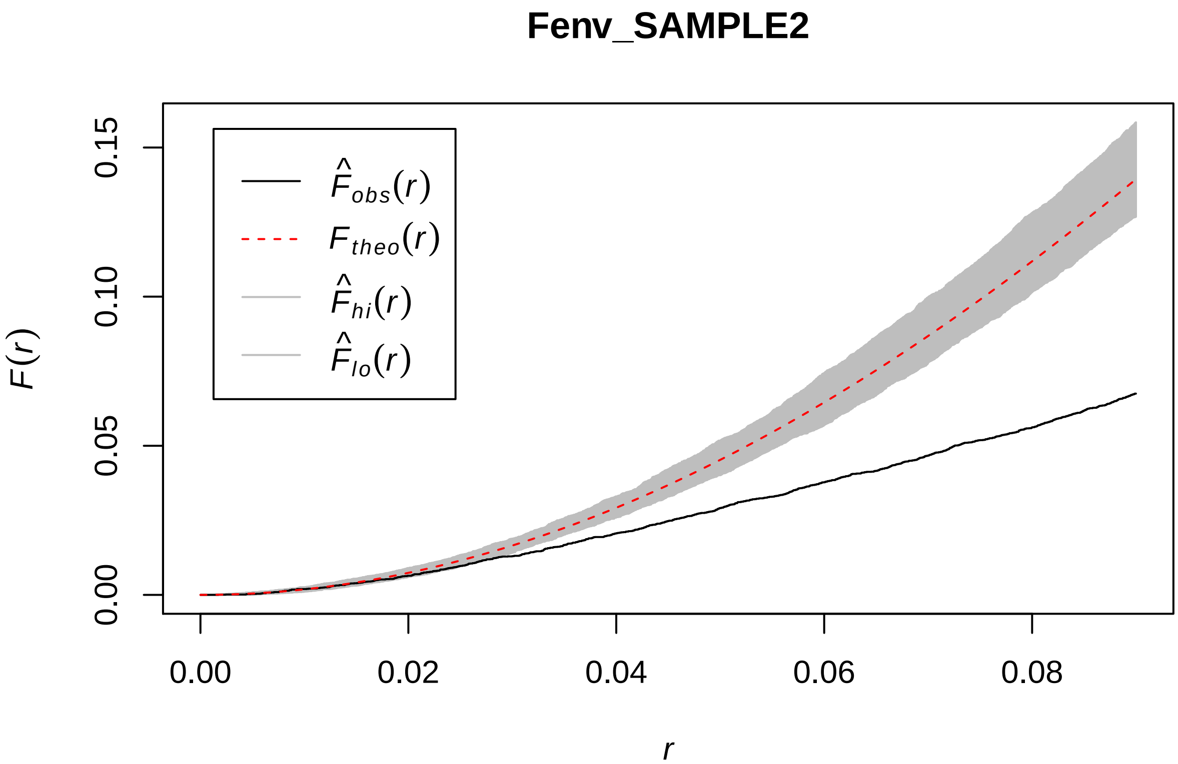
```r
summary(Fenv_SAMPLE2)
```

```
## Pointwise critical envelopes for F(r)
## and observed value for 'ppp(x = sp_point_SAMPLE2[, 1], y =
## sp_point_SAMPLE2[, 2], window = owin(bbox(sp_point_SAMPLE2)[1, '
## Obtained from 99 simulations of CSR
## Alternative: two.sided
## Upper envelope: pointwise 2nd largest of simulated curves
## Lower envelope: pointwise 2nd smallest of simulated curves
## Significance level of Monte Carlo test: 4/100 = 0.04
## Data: ppp(x = sp_point_SAMPLE2[, 1], y = sp_point_SAMPLE2[, 2], window =
## owin(bbox(sp_point_SAMPLE2)[1,
```

```r
plot(Fenv_SAMPLE1)
```




```r
plot(Fenv_SAMPLE2)
```

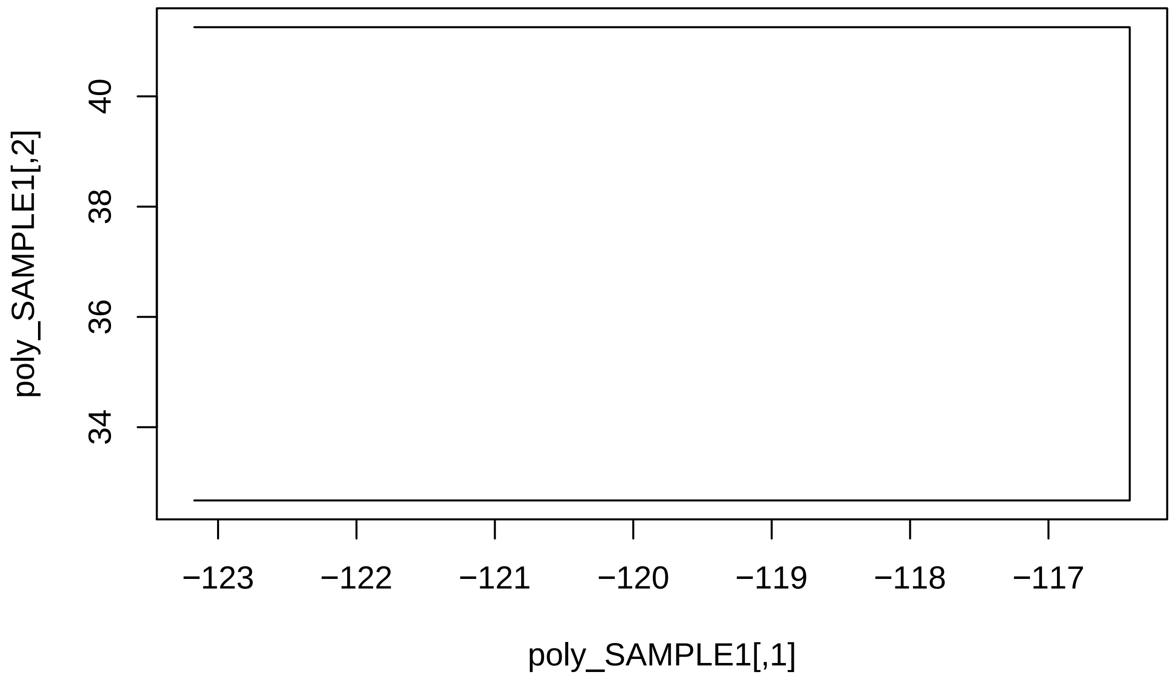


Commonalities: (1) F values from both data SAMPLE1 and data SAMPLE2 are below the envelope of the Fs under Poisson that both samples shows data clustering. (2) simulated curve of pointwise critical envelope is the same for both SAMPLE1 and SAMPLE2.

Differences: in the above plots, (1) the curve for F values of SAMPLE1 is less smooth and higher than SAMPLE2. (2) The lower envelope for SAMPLE1 is lower and the curve is less smooth than the lower envelope of SAMPLE2. The upper envelope for SAMPLE1 is more choppy than SAMPLE2. Thus, the range between upper envelope and lower envelope of SAMPLE1 is bigger than SAMPLE2 which means there is higher variablility in randomazation envelope of SAMPLE1.

Part2: Use SAMPLE1 and SAMPLE2 to build kernel densities and describe differences and commonalities between the two samples.

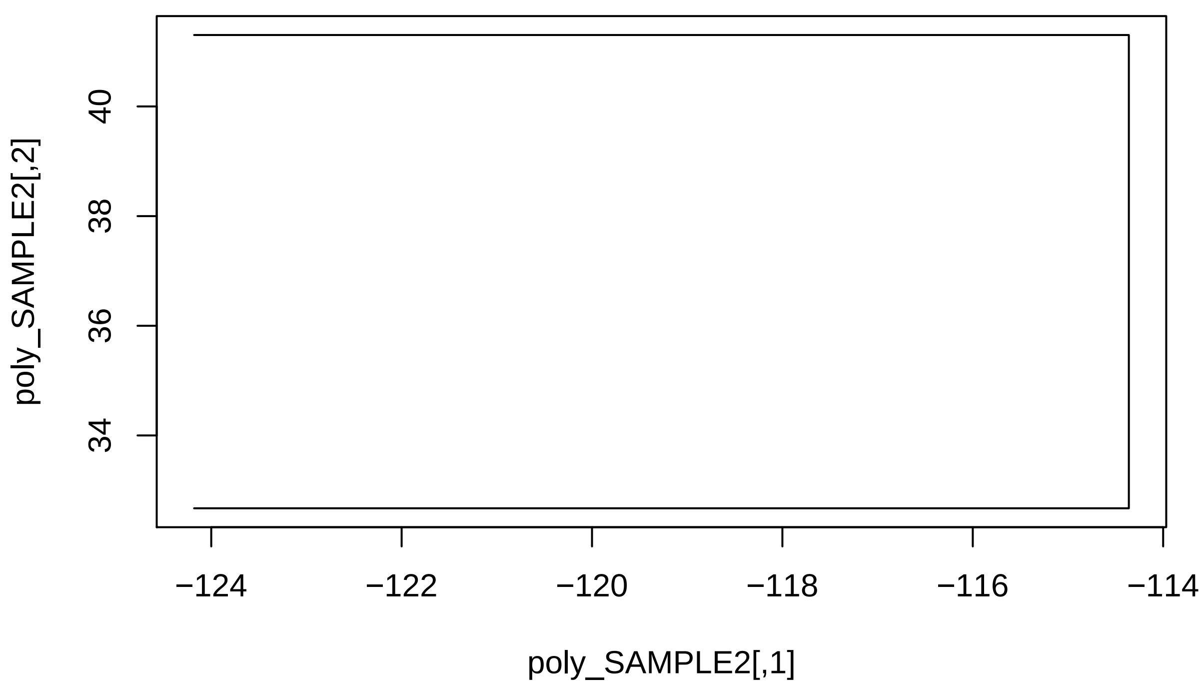
```
poly_SAMPLE1 <- as.points(c(min(sp_point_SAMPLE1[,1]),max(sp_point_SAMPLE1[,1]),
                          max(sp_point_SAMPLE1[,1]),min(sp_point_SAMPLE1[,1])),c(max(sp_point_SAMPLE1[,2]),
plot(poly_SAMPLE1, type="l")
```




```
class(poly_SAMPLE1)
```

```
## [1] "matrix"
```

```
poly_SAMPLE2 <- as.points(c(min(sp_point_SAMPLE2[,1]),max(sp_point_SAMPLE2[,1]),
                max(sp_point_SAMPLE2[,1]),min(sp_point_SAMPLE2[,1])),c(max(sp_point_SAMPLE2[,2]),
plot(poly_SAMPLE2, type="l")
```




```
class(poly_SAMPLE2)
```

```
## [1] "matrix"
```

```r
mserw_SAMPLE1 <- mse2d(sp_point_SAMPLE1, poly=poly_SAMPLE1, nsmse=100, range=0.1)
summary(mserw_SAMPLE1)
```

```
##     Length Class  Mode
## mse 100    -none- numeric
## h   100    -none- numeric
```

```r
class(mserw_SAMPLE1)
```

```
## [1] "list"
```

```r
summary(mserw_SAMPLE1$mse, mser_SAMPLE1w$h)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.11     15.83     59.79   3003.00    261.45 184636.51
```

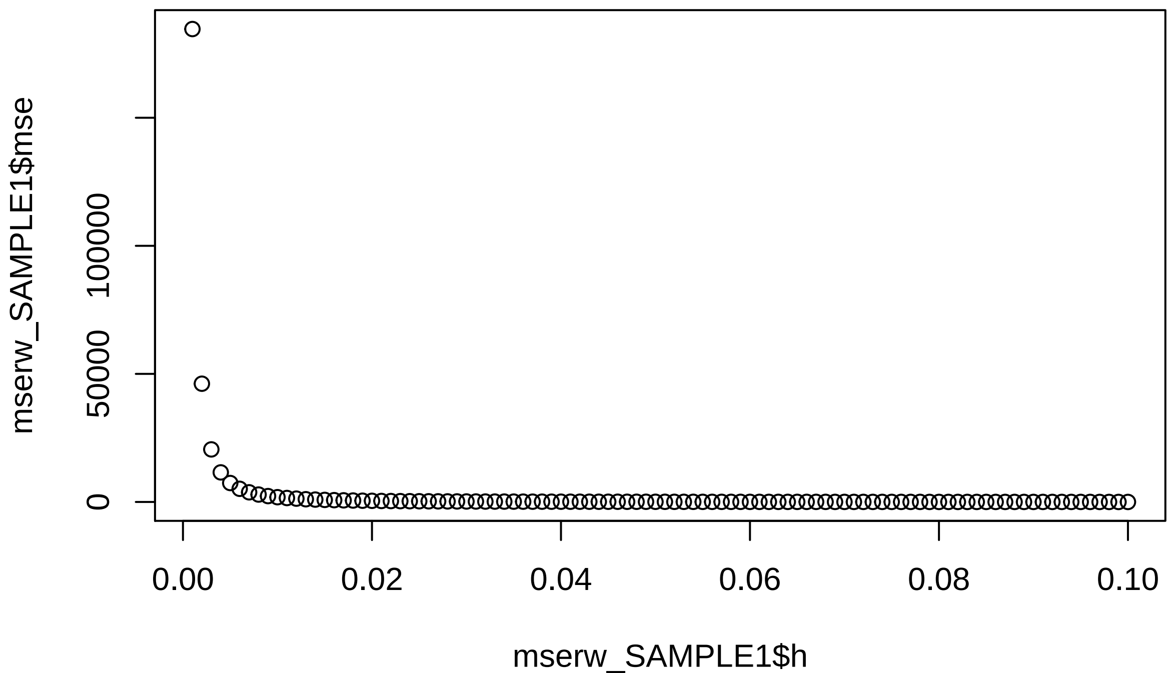
```r
print(mserw_SAMPLE1)
```

```
## $mse
##   [1] 1.846365e+05 4.615913e+04 2.051517e+04 1.153978e+04 7.385460e+03
##   [6] 5.128792e+03 3.769800e+03 2.890408e+03 2.287132e+03 1.855538e+03
##  [11] 1.537055e+03 1.295232e+03 1.063026e+03 9.190945e+02 8.029797e+02
##  [16] 7.080834e+02 6.295641e+02 5.636854e+02 4.874579e+02 4.417424e+02
##  [21] 3.855155e+02 3.526474e+02 3.239238e+02 2.986468e+02 2.763151e+02
##  [26] 2.564972e+02 2.389281e+02 2.138342e+02 2.004033e+02 1.800973e+02
##  [31] 1.697101e+02 1.530811e+02 1.449565e+02 1.375290e+02 1.307059e+02
##  [36] 1.187314e+02 1.132506e+02 1.081865e+02 1.035050e+02 9.916756e+01
##  [41] 9.515380e+01 9.146107e+01 8.804625e+01 8.105572e+01 7.825702e+01
##  [46] 7.562618e+01 7.314980e+01 6.760493e+01 6.552457e+01 6.355911e+01
##  [51] 5.602682e+01 5.175909e+01 4.777304e+01 4.657534e+01 4.300179e+01
##  [56] 4.201347e+01 4.107152e+01 3.797721e+01 3.506962e+01 3.233741e+01
##  [61] 2.976622e+01 2.926241e+01 2.877365e+01 2.469519e+01 2.434767e+01
##  [66] 2.401203e+01 2.369129e+01 2.338054e+01 2.153020e+01 2.128346e+01
##  [71] 1.957712e+01 1.937983e+01 1.779763e+01 1.764207e+01 1.617665e+01
##  [76] 1.478322e+01 1.470212e+01 1.340609e+01 1.217113e+01 1.099659e+01
##  [81] 8.752932e+00 5.518633e+00 4.581268e+00 3.686080e+00 3.854847e+00
##  [86] 3.017925e+00 2.218797e+00 2.410692e+00 2.595458e+00 2.773318e+00
##  [91] 2.944389e+00 2.236467e+00 2.414741e+00 2.588177e+00 2.755245e+00
##  [96] 2.114354e+00 1.500049e+00 1.680158e+00 6.576636e-01 1.107084e-01
##
## $h
##   [1] 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 0.010 0.011
##  [12] 0.012 0.013 0.014 0.015 0.016 0.017 0.018 0.019 0.020 0.021 0.022
##  [23] 0.023 0.024 0.025 0.026 0.027 0.028 0.029 0.030 0.031 0.032 0.033
##  [34] 0.034 0.035 0.036 0.037 0.038 0.039 0.040 0.041 0.042 0.043 0.044
##  [45] 0.045 0.046 0.047 0.048 0.049 0.050 0.051 0.052 0.053 0.054 0.055
##  [56] 0.056 0.057 0.058 0.059 0.060 0.061 0.062 0.063 0.064 0.065 0.066
##  [67] 0.067 0.068 0.069 0.070 0.071 0.072 0.073 0.074 0.075 0.076 0.077
##  [78] 0.078 0.079 0.080 0.081 0.082 0.083 0.084 0.085 0.086 0.087 0.088
##  [89] 0.089 0.090 0.091 0.092 0.093 0.094 0.095 0.096 0.097 0.098 0.099
## [100] 0.100
```

```r
help(mse2d_SAMPLE1)
```

```
## No documentation for 'mse2d_SAMPLE1' in specified packages and libraries:
## you could try '??mse2d_SAMPLE1'
```

```r
plot(mserw_SAMPLE1$h, mserw_SAMPLE1$mse)
```




```r
mserw_SAMPLE2 <- mse2d(sp_point_SAMPLE2, poly=poly_SAMPLE2, nsmse=100, range=0.1)
summary(mserw_SAMPLE2)
```

```
##     Length Class  Mode
## mse 100    -none- numeric
## h   100    -none- numeric
```

```r
class(mserw_SAMPLE2)
```

```
## [1] "list"
```

```r
summary(mserw_SAMPLE2$mse, mser_SAMPLE2w$h)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   -21.13   -20.27   -13.90   846.45    41.34 53952.83
```

```r
print(mserw_SAMPLE2)
```

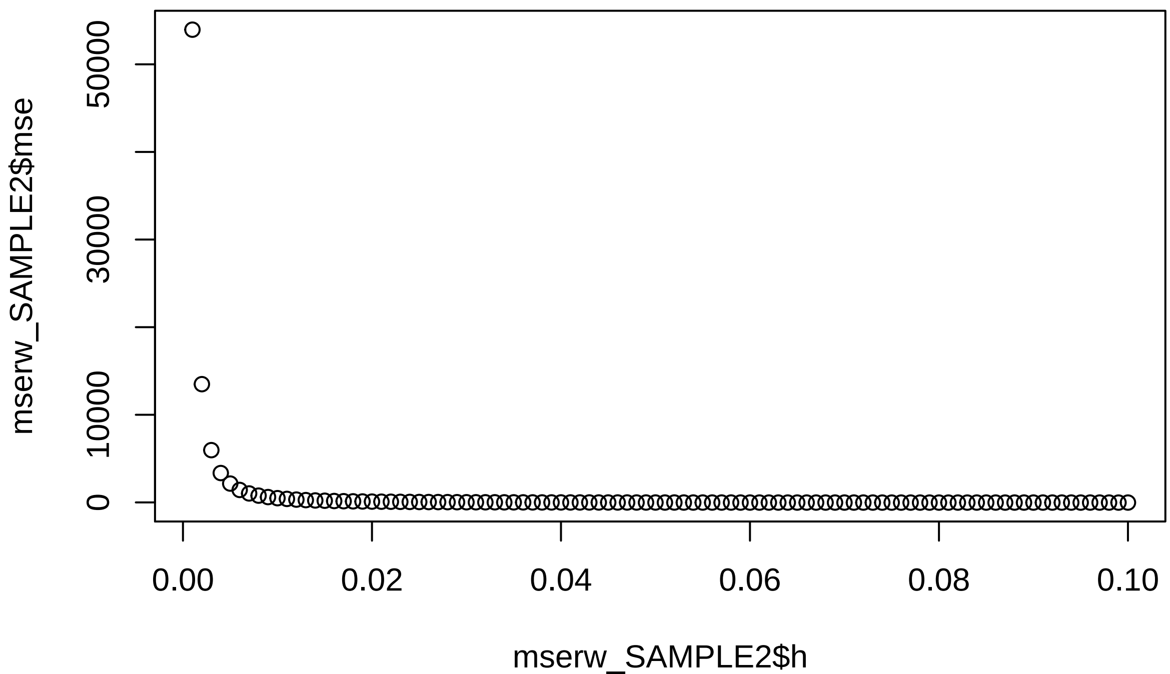
```
## $mse
##   [1] 53952.826766 13495.991734  5963.339010  3350.508438  2146.478981
##   [6]  1420.875806  1020.703687   771.463818   607.845831   482.714879
##  [11]   398.350860   319.398614   259.462082   221.931413   192.091221
##  [16]   156.283334   136.478561   120.152873   105.383377    91.957441
##  [21]    82.596069    74.592477    62.814116    55.017925    45.539929
##  [26]    39.937747    35.078928    28.106690    23.055351    17.203092
##  [31]    13.879790    11.805290    10.374179     6.496538     3.758374
##  [36]     2.662524    -2.080478    -2.721477    -2.705351    -4.826798
##  [41]    -5.957244    -6.721349    -7.399537    -8.796871    -9.931471
##  [46]   -10.952736   -11.285274   -12.505861   -12.529653   -13.395817
##  [51]   -14.674440   -14.402880   -15.978784   -16.467587   -17.115538
##  [56]   -16.188660   -16.914118   -17.447298   -18.040991   -18.101773
##  [61]   -18.839417   -18.729399   -18.503591   -19.029617   -18.985398
##  [66]   -19.327840   -18.676256   -18.891510   -19.082786   -19.339584
```

```
##  [71]   -19.571991   -20.031661   -20.457586   -20.694238   -20.908939
##  [76]   -20.879009   -21.132894   -21.081201   -21.037245   -20.839365
##  [81]   -20.708256   -20.446494   -20.693847   -20.368366   -20.348970
##  [86]   -20.325633   -20.638767   -20.484785   -20.381525   -20.281663
##  [91]   -20.069877   -20.145842   -20.262006   -20.581880   -20.786065
##  [96]   -20.554088   -20.566379   -20.516695   -20.689097   -20.487438
##
## $h
##   [1] 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 0.010 0.011
##  [12] 0.012 0.013 0.014 0.015 0.016 0.017 0.018 0.019 0.020 0.021 0.022
##  [23] 0.023 0.024 0.025 0.026 0.027 0.028 0.029 0.030 0.031 0.032 0.033
##  [34] 0.034 0.035 0.036 0.037 0.038 0.039 0.040 0.041 0.042 0.043 0.044
##  [45] 0.045 0.046 0.047 0.048 0.049 0.050 0.051 0.052 0.053 0.054 0.055
##  [56] 0.056 0.057 0.058 0.059 0.060 0.061 0.062 0.063 0.064 0.065 0.066
##  [67] 0.067 0.068 0.069 0.070 0.071 0.072 0.073 0.074 0.075 0.076 0.077
##  [78] 0.078 0.079 0.080 0.081 0.082 0.083 0.084 0.085 0.086 0.087 0.088
##  [89] 0.089 0.090 0.091 0.092 0.093 0.094 0.095 0.096 0.097 0.098 0.099
## [100] 0.100
```

```r
help(mse2d_SAMPLE2)
```

```
## No documentation for 'mse2d_SAMPLE2' in specified packages and libraries:
## you could try '??mse2d_SAMPLE2'
```

```r
plot(mserw_SAMPLE2$h, mserw_SAMPLE2$mse)
```



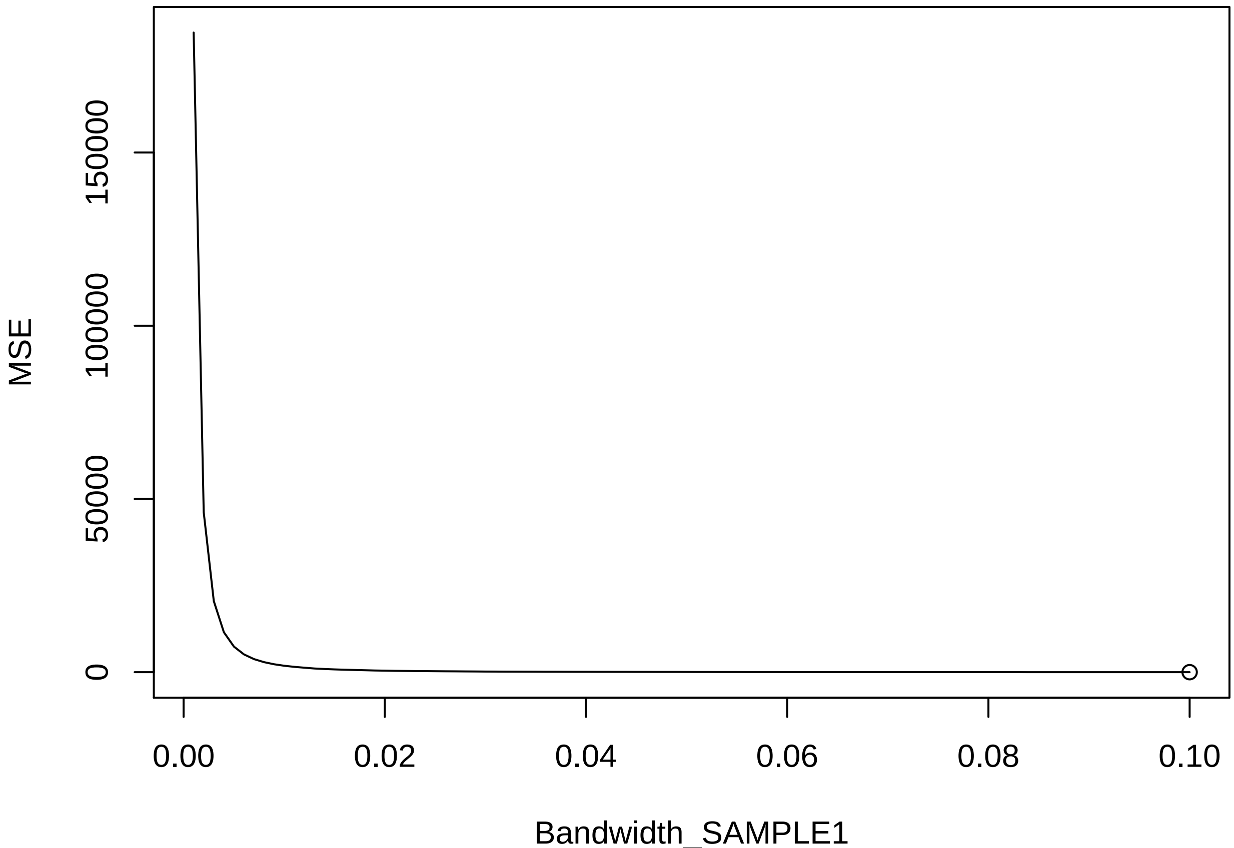

```r
bw_SAMPLE1 <- mserw_SAMPLE1$h[which.min(mserw_SAMPLE1$mse)] ## Bandwidth=.01
summary(bw_SAMPLE1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.1     0.1     0.1     0.1     0.1     0.1
```

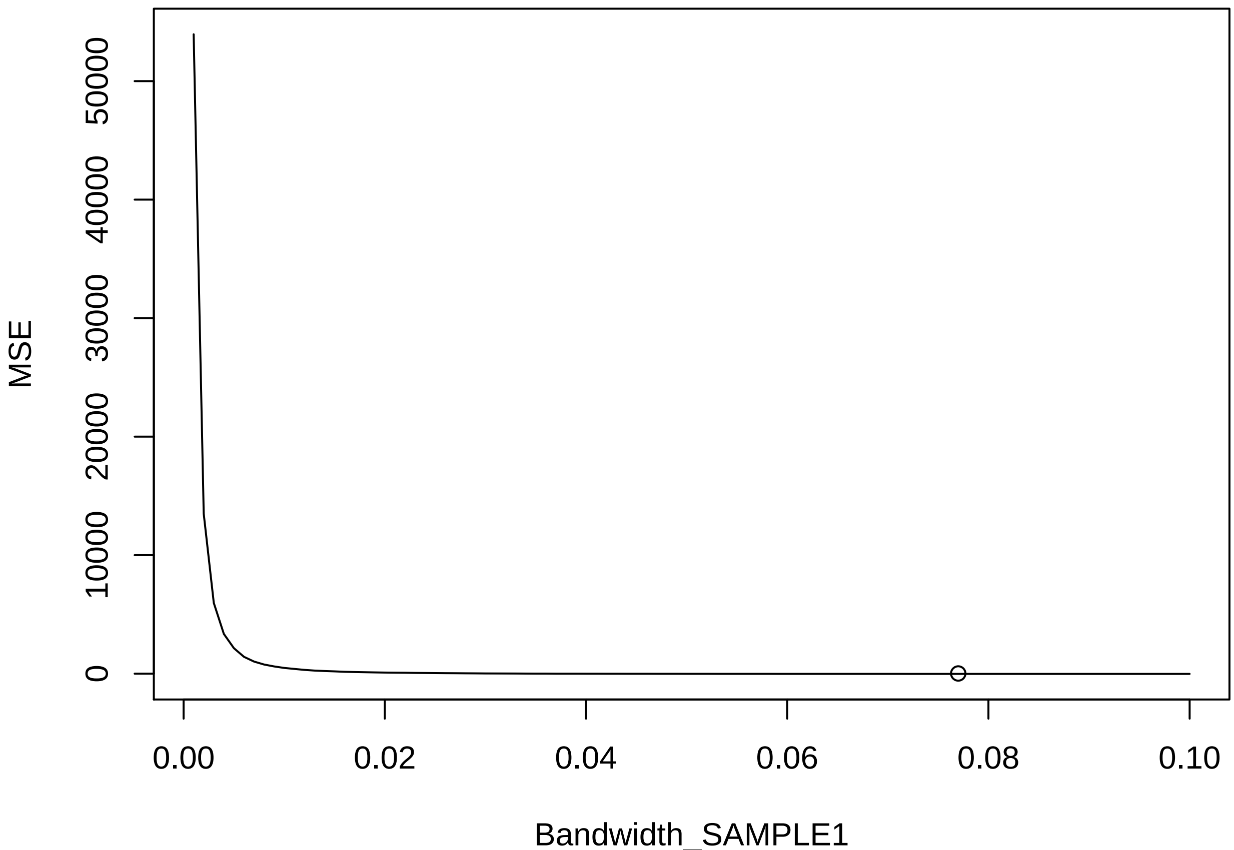
```r
bw_SAMPLE2 <- mserw_SAMPLE2$h[which.min(mserw_SAMPLE2$mse)] ## Bandwidth=.01
summary(bw_SAMPLE2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.077   0.077   0.077   0.077   0.077   0.077
```

```
par(mar=c(4,4,0.5,0.5))
plot(x=mserw_SAMPLE1$h, y=mserw_SAMPLE1$mse, xlab="Bandwidth_SAMPLE1", ylab="MSE", type="l")
i<-which.min(mserw_SAMPLE1$mse)
points(mserw_SAMPLE1$h[i], mserw_SAMPLE1$mse[i])
```




```
plot(x=mserw_SAMPLE2$h, y=mserw_SAMPLE2$mse, xlab="Bandwidth_SAMPLE1", ylab="MSE", type="l")
j<-which.min(mserw_SAMPLE2$mse)
points(mserw_SAMPLE1$h[j], mserw_SAMPLE1$mse[j])
```




```
sp_points_SAMPLE1 <- SpatialPoints(coords=sp_point_SAMPLE1, proj4string=CRS("+proj=utm +zone=
10 +datum=WGS84"))
class(sp_points_SAMPLE1)
```

```
## [1] "SpatialPoints"
## attr(,"package")
## [1] "sp"
```

```
sp_points_SAMPLE2 <- SpatialPoints(coords=sp_point_SAMPLE2, proj4string=CRS("+proj=utm +zone=
10 +datum=WGS84"))
class(sp_points_SAMPLE2)
```

```
## [1] "SpatialPoints"
## attr(,"package")
## [1] "sp"
```

```
grd_SAMPLE1 <- Sobj_SpatialGrid(sp_points_SAMPLE1,maxDim=100)$SG
class(grd_SAMPLE1)
```

```
## [1] "SpatialGrid"
## attr(,"package")
## [1] "sp"
```

```
grd_SAMPLE2 <- Sobj_SpatialGrid(sp_points_SAMPLE2,maxDim=100)$SG
class(grd_SAMPLE2)
```

```
## [1] "SpatialGrid"
## attr(,"package")
## [1] "sp"
```

```
grd_SAMPLE1 <- GridTopology(summary(grd_SAMPLE1)$grid[,1],
                           cellsize=summary(grd_SAMPLE1)$grid[,2],
                           cells.dim=summary(grd_SAMPLE1)$grid[,3])
class(grd_SAMPLE1)
```

```
## [1] "GridTopology"
## attr(,"package")
## [1] "sp"
```

```
grd_SAMPLE2 <- GridTopology(summary(grd_SAMPLE2)$grid[,1],
                           cellsize=summary(grd_SAMPLE2)$grid[,2],
                           cells.dim=summary(grd_SAMPLE2)$grid[,3])
class(grd_SAMPLE2)
```

```
## [1] "GridTopology"
## attr(,"package")
## [1] "sp"
```

I estimate kernel density based on minimizing mean squared error with an equation derived by Diggle.

```
kernel_SAMPLE1 <- spkernel2d(sp_point_SAMPLE1, poly=poly_SAMPLE1, h0=bw_SAMPLE1, grd=grd_SAMPLE1)
class(kernel_SAMPLE1)
```

```
## [1] "numeric"
```

```
summary(kernel_SAMPLE1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   1.699   0.000 159.930
```

```
kernel_SAMPLE2 <- spkernel2d(sp_point_SAMPLE2, poly=poly_SAMPLE2, h0=bw_SAMPLE2, grd=grd_SAMPLE2)
class(kernel_SAMPLE2)
```

```
## [1] "numeric"
```

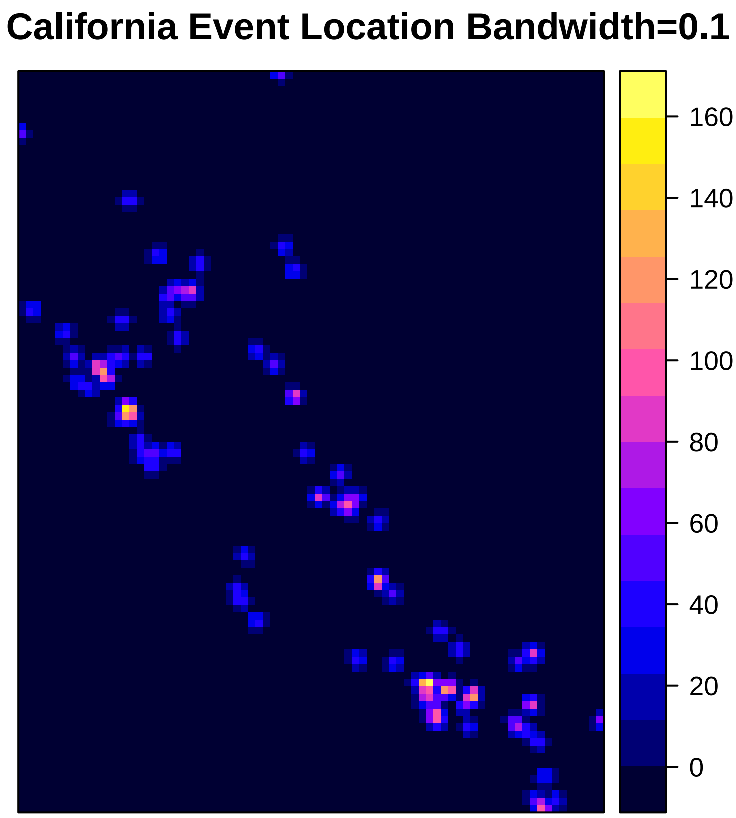
```
summary(kernel_SAMPLE2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   5.901   0.000 679.509
```

```
help(spkernel2d)
```

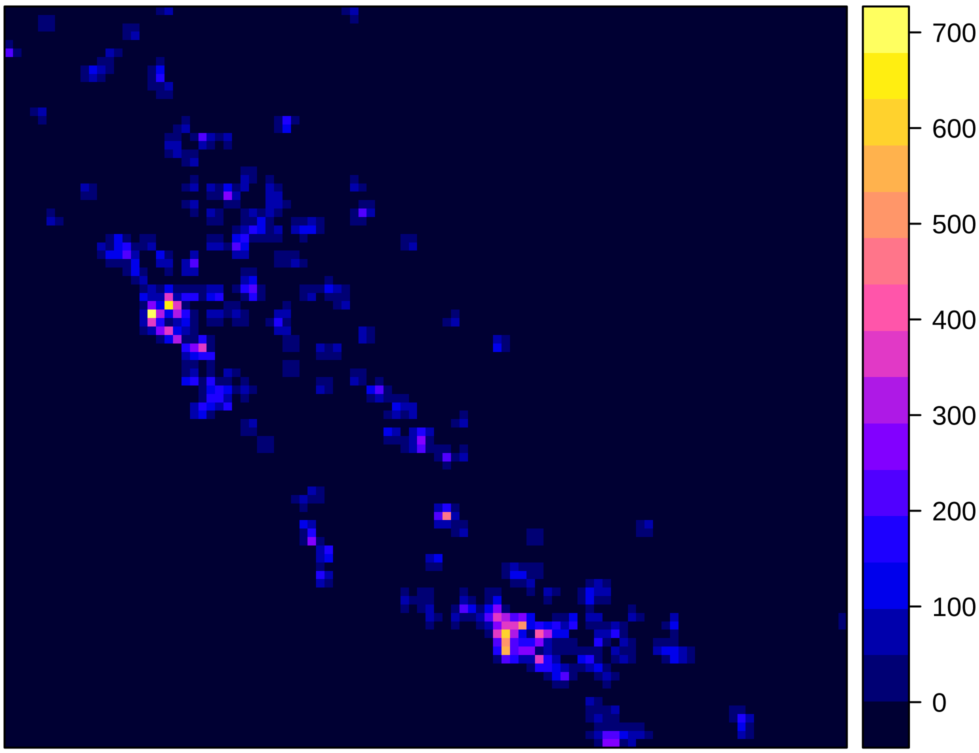
Saving the kernel deinsity estimates in one database and a grided databse (SpatialGridDataFrame) Then, plot the kernel densities. Observe the kernel density of the optimized bandwidth h0=0.1 for SAMPLE1 and h0=0.077 for SAMPLE2.

```
CAdf_SAMPLE1 <- data.frame(kernel1=kernel_SAMPLE1)
CAsg_SAMPLE1 <- SpatialGridDataFrame(grd_SAMPLE1, data=CAdf_SAMPLE1)
spplot(CAsg_SAMPLE1, main="California Event Location Bandwidth=0.1")
```




```r
CAdf_SAMPLE2 <- data.frame(kernel1=kernel_SAMPLE2)
CAsg_SAMPLE2 <- SpatialGridDataFrame(grd_SAMPLE2, data=CAdf_SAMPLE2)
spplot(CAsg_SAMPLE2, main="California Event Location Bandwidth=0.077")
```

California Event Location Bandwidth=0.077



Commonality: optimized bandwidths for SAMPLE1 and SAMPLE2 get similar kernel density across California event locations (similar kernel density intensity ditribution). **Differences:** (1) Although SAMPLE1 and SAMPLE2 come from the same dataset, they have different optimized bandwidths. SAMPLE1 (smaller sample size) has bigger optimized bandwidth while SAMPLE2 (bigger sample size) has smaller optimized bandwidth. (2) The kernel density shows more California event locations for bigger sample size (SAMPLE2) than smaller sample size (SAMPLE1).

Part 3:
(1)Compare the Kriging examples in Isaacs and Srivastava and the class notes of the week of March 5 ??? describe differences.

First, the Kriging examples in Isaacs and Srivastava use a covariance function while gstat use variogram. Second, the Kriging examples in Isaacs and Srivastava impose 7 points to the data while gstat fit a nonlinear equation to the data.

(2)Then, compare them to the examples in ???Phaedon???s Examples of variograms??? document on Gaucho space. What type of spatial distribution and continuity does the Isaacs&Srivastava represent and how is that different from the data analyzed in the Lecture Notes of March 6?

Isaacs&Srivastava represents discontinous near origin semivariogram shape, which is highly irregular (quasi-random) spatial variability at small scales. The data analyzed in the Lecture Notes of March 6 represents linear shape near origin semivariogram shape, which has continuous spatial variability (not extremely smooth) and spatial variables are not differentiable.