Assignment 1

Manan Lohia (e0998061)

2022-09-20

Q1) The R object in `abc.rds` contains the number of births (BIR74) and birth defects (SID74) in 100 counties of a state of US for the period July 1, 1974 to June 30, 1978.

a) What type of spatial object is `abc` ?

```
class(abc)
```

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

`abc` is a **SpatialPolygonsDataFrame**

b) Find the CRS of `abc`

```
proj4string(abc)
```

```
## [1] "+proj=longlat +datum=NAD27 +no_defs"
```

`abc` uses the `NAD27` CRS.

c) What commands in R transform the coordinate system of abc to WGS84? Specify the exact code.

The `spTransform` function allows us to transform the CRS of datasets

```
abc_wgs84 <- spTransform(abc, CRS("+proj=longlat +datum=WGS84"))
proj4string(abc_wgs84)
```

```
## [1] "+proj=longlat +datum=WGS84 +no_defs"
```

d) How would you compute the rate of birth defects per 10,000 births over the four year period?

To calculate the rate of birth defects per 10,000 over the 4 year period, we calculate the rate of birth defects and multiply it by 10,000

```
abc$rate <- 10000 * abc$SID74/abc$BIR74

head(slot(abc, 'data'))
```

```
##       SP_ID CNTY_ID east north L_id M_id     names  AREA PERIMETER CNTY_
## 37001 37001    1904  278   151    1    3  Alamance 0.111     1.392  1904
## 37003 37003    1950  179   142    2    2 Alexander 0.066     1.070  1950
## 37005 37005    1827  183   182    1    2 Alleghany 0.061     1.231  1827
## 37007 37007    2096  240    75    3    2     Anson 0.138     1.621  2096
## 37009 37009    1825  164   176    1    2      Ashe 0.114     1.442  1825
## 37011 37011    1892  138   154    1    2     Avery 0.064     1.213  1892
##            NAME  FIPS FIPSNO CRESS_ID BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79
## 37001  Alamance 37001  37001        1  4672    13    1243  5767    11    1397
## 37003 Alexander 37003  37003        2  1333     0     128  1683     2     150
## 37005 Alleghany 37005  37005        3   487     0      10   542     3      12
## 37007     Anson 37007  37007        4  1570    NA     952  1875     4    1161
## 37009      Ashe 37009  37009        5  1091     1      10  1364     0      19
## 37011     Avery 37011  37011        6   781     0       4   977     0       5
##          raw.74       EB       DA      rate
## 37001  5.565068 5.017129  3.870963 27.825342
## 37003  0.000000 2.682216  1.858859  0.000000
## 37005  0.000000 3.410755  2.010978  0.000000
## 37007 19.108280 9.677608 10.261004        NA
## 37009  1.833181 3.394595  1.734804  9.165903
## 37011  0.000000 3.116575  1.472093  0.000000
```

e) How about the rate of birth defects per 10,000 births in a typical year during the period?

To estimate the rate of birth defects per 10,000 for a typical year, we assume that the rate of birth and rate of birth defects are both constant across the four years, as we do not have any information about the trend of the rates over the 4 years. Hence, this would involve dividing the numerator and the denominator by the same factor, leading us to the same value

f) The entry for SID74 is "NA" for one of the counties. Change it to 0 so that it does not cause problems in the computations below. Take note of the name of this county.

```
abc@data[is.na(abc$SID74),]
```

```
##       SP_ID CNTY_ID east north L_id M_id names  AREA PERIMETER CNTY_  NAME
## 37007 37007    2096  240    75    3    2 Anson 0.138     1.621  2096 Anson
##       FIPS FIPSNO CRESS_ID BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79   raw.74
## 37007 37007  37007        4  1570    NA     952  1875     4    1161 19.10828
##            EB     DA rate
## 37007 9.677608 10.261   NA
```

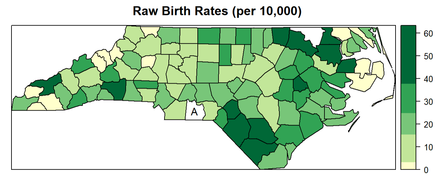
The name of the which had the missing value "NA" is **Anson**

```
abc@data$SID74[is.na(abc@data$SID74)] = 0
abc@data[is.na(abc$SID74),]
```

```
##  [1] SP_ID     CNTY_ID   east      north     L_id      M_id      names
##  [8] AREA      PERIMETER CNTY_     NAME      FIPS      FIPSNO    CRESS_ID
## [15] BIR74     SID74     NWBIR74   BIR79     SID79     NWBIR79   raw.74
## [22] EB        DA        rate
## <0 rows> (or 0-length row.names)
```

g) Provide a choropleth map of the raw rates per 10,000 births over the four year period. The map should be simple and visually appealing.Mark the county which you made the change in (f) with the letter A on the county in the map.

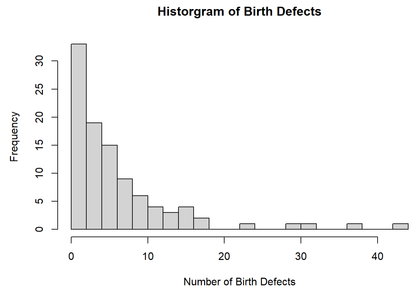
```
q5a <- classIntervals(abc$rate, n=5, "fisher")
pal = brewer.pal(5, 'YlGn')
spplot(abc, "rate", at=q5a$brks, col.regions=pal, main="Raw Birth Rates (per 10,000)",
       sp.layout=list("sp.text", coordinates(abc[abc$NAME == 'Anson',]), c('A')))
```



h) Are there counties with very few birth defect counts? What is the problem with displaying the raw rates for such counties? What is the commonly referred to name for this problem?

```
hist(abc$SID74, main = 'Historgram of Births', xlab = 'Number of Births', breaks = 20)
```


```
hist(abc$SID74, main = 'Historgram of Birth Defects', xlab = 'Number of Birth Defects', breaks = 20)
```



We observe that there exist counties which have only a handful of birth defects, relative to other counties. The problem with displaying the raw rates in this case is that we run the risk of getting a large value for the calculated rate due to a small population (small number of births) despite the small number of birth defects. This skews the perception and can lead to flawed inferences.

This problem is often termed as the **small number problem**

i) One way of dealing with the problem in (h) is to group counties so that the groups have roughly equal population sizes. Comment on the advantages and disadvantages of doing so.

The benefit of grouping the counties is that we are able to distribution of population sizes uniform. Since each group has a similar population size, the number of defects in the region would be a better estimator of the true rate in the region.

However, the flipside of performing such a grouping is that we will lose some information regarding the distribution of birth and birth defects within the regions that were grouped together. This loss of information is not always desirable.

1. j We have,

$$Y_i \sim \text{Poisson}(n_i \xi_i) \Rightarrow \text{Var}(Y_i/n_i) = \xi_i/n_i$$

~~$\Rightarrow \text{Var}(\hat{\xi}_i) = \xi_i/n_i$~~

$$\text{Var}(\hat{\xi}_i) = \text{Var}\left[\frac{\sum_j w_{ij} Y_j}{\sum_j w_{ij} n_j}\right]$$

$$= \frac{\sum w_{ij}^2 \text{Var}(Y_j)}{(\sum w_{ij} n_j)^2}$$

Here, $w_{ij} = e^{-\frac{d_{ij}^2}{1000}} \Rightarrow 0 < w_{ij} \leq 1$

$\Rightarrow w_{ij}^2 \leq w_{ij}$

~~[illegible]~~

$$\therefore \text{Var}(\hat{\xi}_i) = \frac{\sum w_{ij}^2 \text{Var}(Y_j)}{(\sum w_{ij} n_j)^2}$$

$$\leq \frac{\sum w_{ij} \text{Var}(Y_j)}{(\sum w_{ij} n_j)^2}$$

$$= \frac{\sum w_{ij} n_j \xi}{(\sum w_{ij} n_j)^2} = \frac{\xi}{\sum w_{ij} n_j}$$

$$\therefore \text{Var}(\hat{\xi}_i) \leq \frac{\xi_i}{\sum_j w_{ij} n_j}$$

Furthermore, $w_{ij} > 0 \quad \forall i, j$

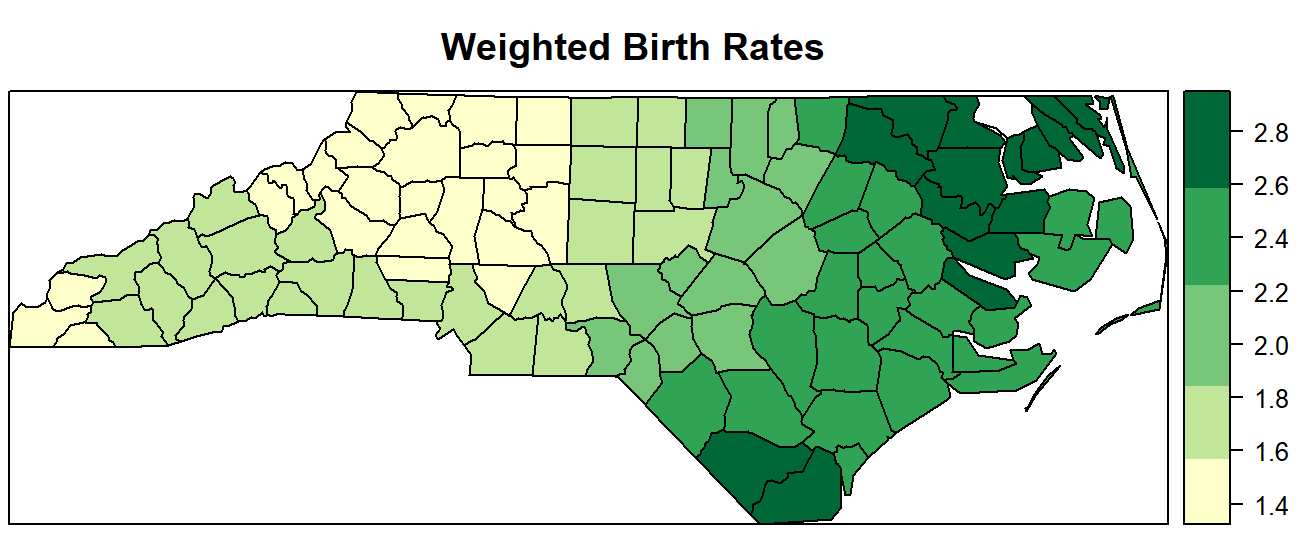
$$\Rightarrow \sum w_{ij} n_j > n_i \quad \forall i \quad [\because w_{ij} = 1 \text{ for } j = i]$$

$$\therefore \frac{\xi_i}{\sum w_{ij} n_j} < \frac{\xi_i}{n_i}$$

$$\therefore \text{Var}(\hat{\xi}_i) \leq \text{Var}(y_i / n_i)$$

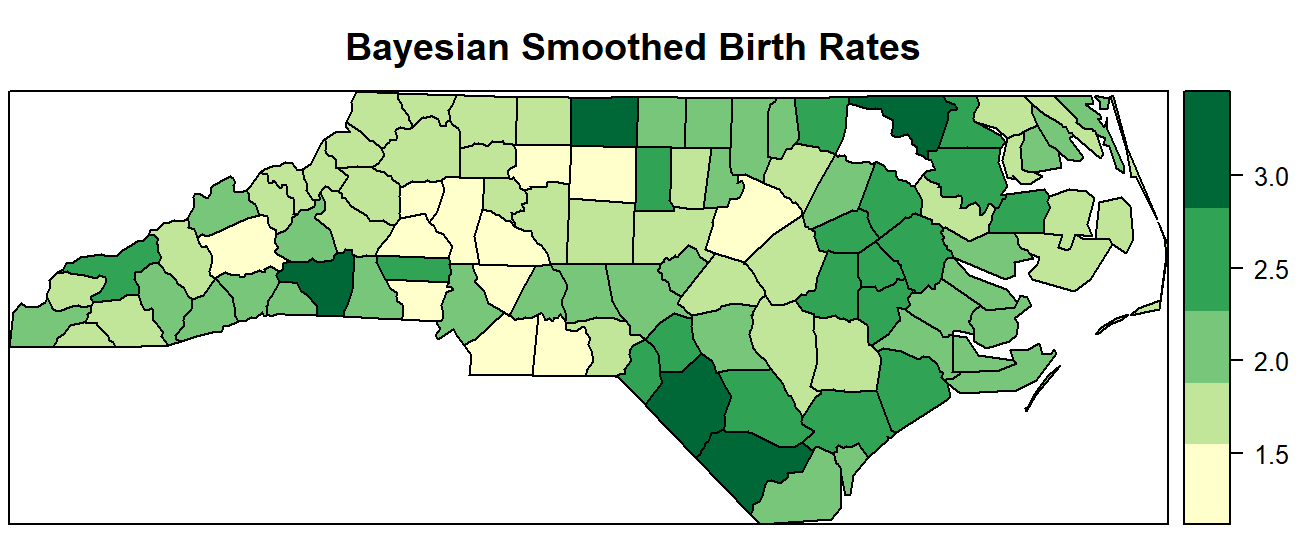
k) Provide a choropleth map of the smoothed rates using the ratio smoother and weights in (j).

```
dists <- spDists(coordinates(abc), longlat = TRUE)
weights <- exp(-(dists**2)/10000)
abc$rate_weighted <- rep(0, times = nrow(weights))

#I multiply the weights by 1000 to improve readability and avoid very small numbers

for (i in 1:nrow(weights))
    abc$rate_weighted[i] <- (sum(weights[i,] * abc$SID74)/sum(weights[i,] * abc$BIR74)) * 1000

q5a <- classIntervals(abc$rate_weighted, n=5, "fisher")
pal = brewer.pal(5, 'YlGn')
spplot(abc, "rate_weighted", at=q5a$brks, col.regions=pal, main="Weighted Birth Rates")
```



l) Provide a choropleth map using the empirical Bayes estimator.

```
# I multiply the estimates by 1000 to improve readability and avoid very small numbers
abc$rate_bayesian <- EBest(abc$SID74, abc$BIR74)$estmm * 1000

q5a <- classIntervals(abc$rate_bayesian, n=5, "fisher")
pal = brewer.pal(5, 'YlGn')
spplot(abc, "rate_bayesian", at=q5a$brks, col.regions=pal, main="Bayesian Smoothed Birth Rates")
```



m) Discuss some of the differences between the ratio smoother and the empirical Bayes estimator.

The major difference between the ratio smoother and the empirical Bayes estimator is that while the ratio smoother takes into consideration the distribution of cases/rates across the population and all the regions, the Bayes estimator only considers the rate of the region under consideration and the global rate. This is evident in the formulas, where Bayes estimator identifies the rate as a population weighted mean of the local and global rate, the ratio smoother calculates the rate as a distribution-based weighted average of the entire population, rather than just one region.

As a result of this, areas with small populations are assigned a rate closer to the global rate through the Bayes estimator, which helps to reduce the impact of the small number problem, but also fails to accurately depict any spatial trends that might be occurring, relative to the ratio smoother. Due to the ratio smoothing method using the population and case counts in all the regions, it does a better job of capturing spatial trends.

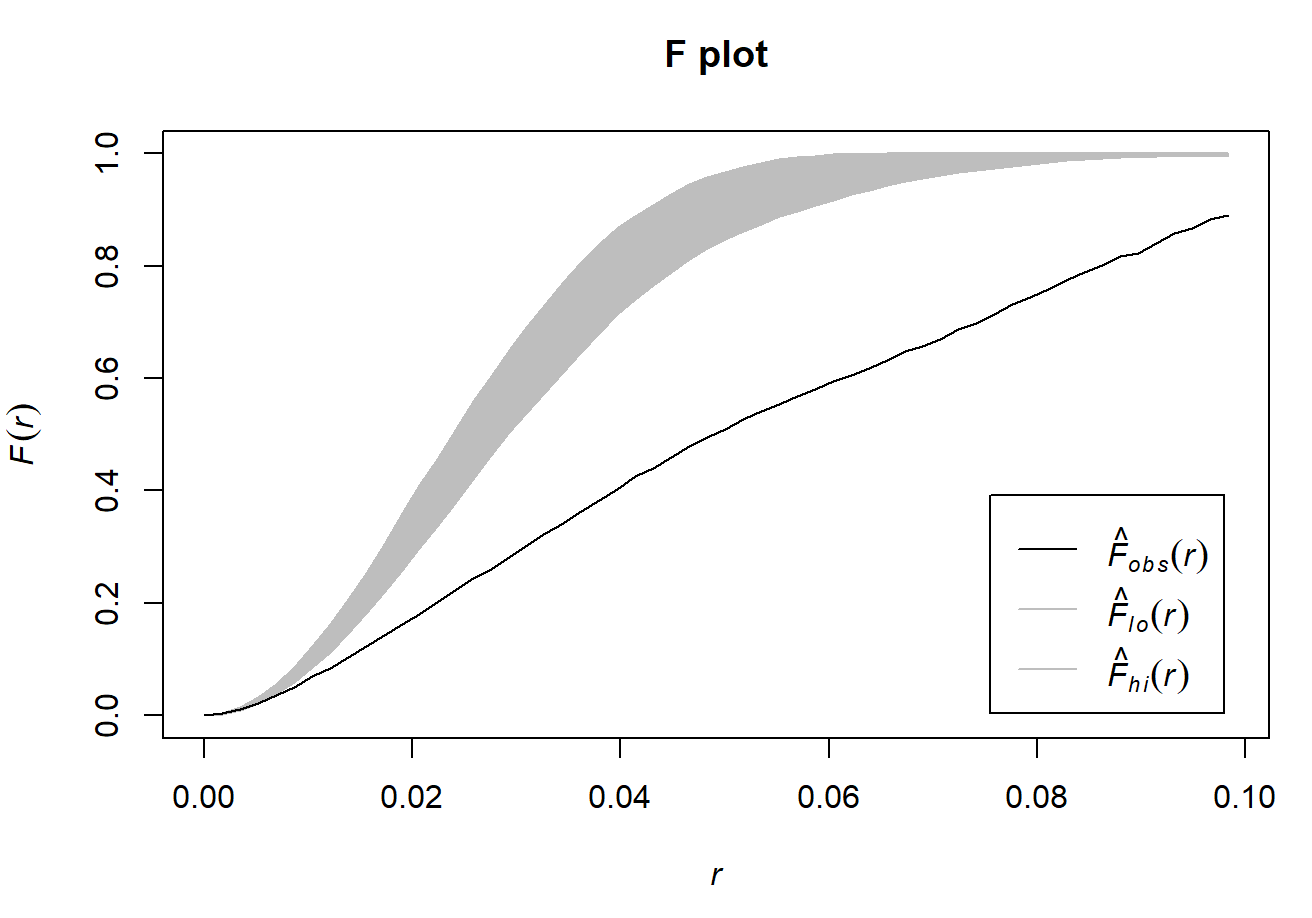
Q2) The R object in `def.rds` is a ppp object containing the locations of individuals with a certain health condition. The R object in `dist.rds` is a list contains two objects d1 and d2. These objects are images containing the distances to two different sources of pollution: Source 1 and Source 2 respectively. Of interest is whether these distances are predictors of the intensity function.

a) Display a plot of the F function of the point pattern against radius *r* (careful!). Set the envelope so that 199 datasets are simulated and the envelope corresponds to 95% pointwise confidence intervals. Write the code so that the same envelope is generated each time the code is run.

```
set.seed(998051)
defenv <- envelope(def, Fest, nrank = 5, nsim = 199)
```

```
## Generating 199 simulations of CSR  ...
## 1, 2, 3, 4.6.8.10.12.14.16.18.20.22.24.26.28.30.32.34.36.38.40
## .42.44.46.48.50.52.54.56.58.60.62.64.66.68.70.72.74.76.78.80
## .82.84.86.88.90.92.94.96.98.100.102.104.106.108.110.112.114.116.118.120
## .122.124.126.128.130.132.134.136.138.140.142.144.146.148.150.152.154.156.158.160
## .162.164.166.168.170.172.174.176.178.180.182.184.186.188.190.192.194.196.198 199.
##
## Done.
```

```
plot(defenv, obs~r, main = 'F plot')
```



b) Write down the null and alternative hypotheses of the test corresponding to $r = 0.06$ and say whether the null hypothesis is rejected.

Under the null hypothesis, we assume that the Point Process exhibits CSR. For $r = 0.06$, the value of the F function under the null hypothesis is $1 - e^{-\lambda\pi(0.06)2}$

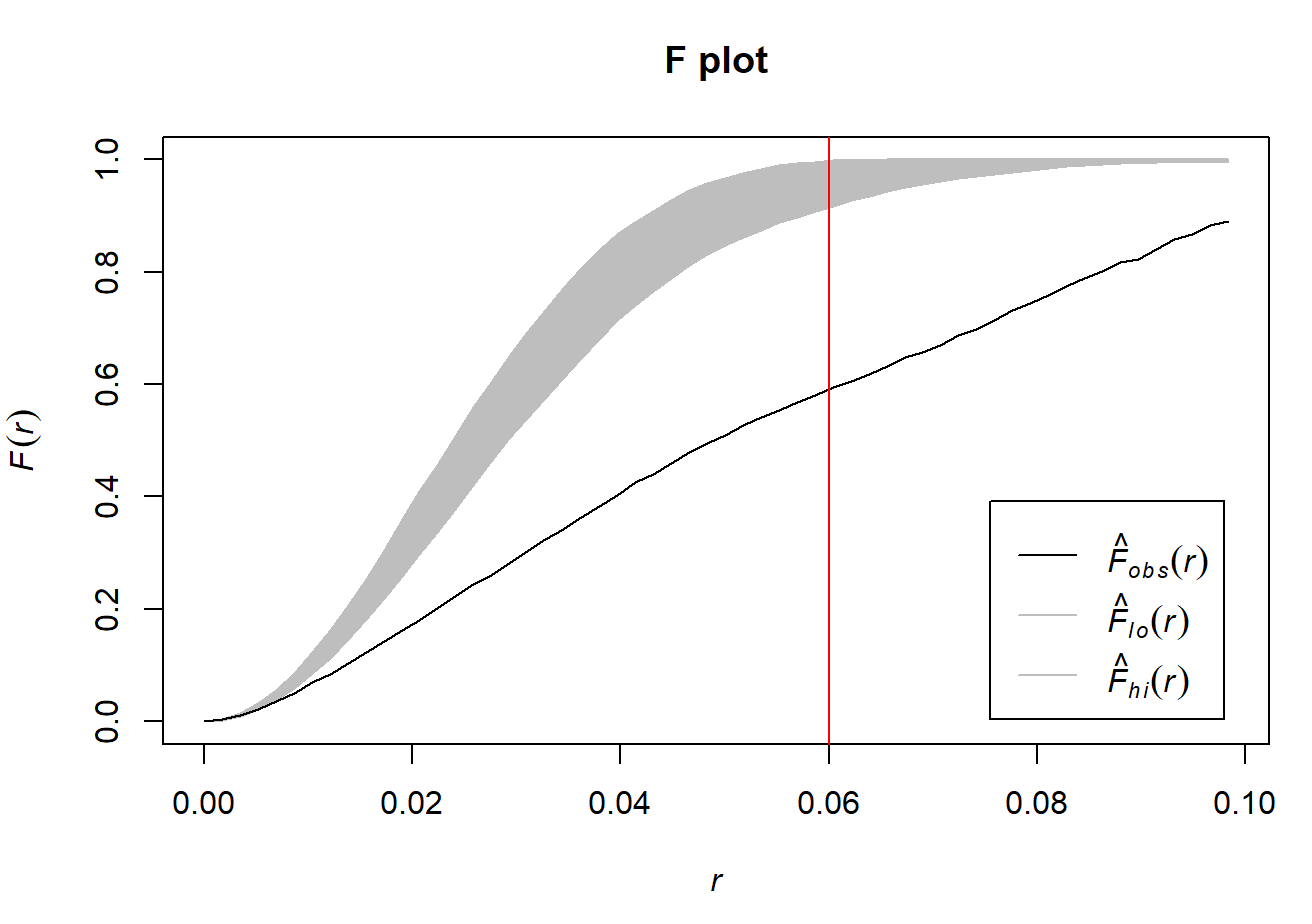
> H_0: $F(0.06) = 1 - e^{-\lambda\pi(0.06)2}$

> H_1: $F(0.06) != 1 - e^{-\lambda\pi(0.06)2}$

```
set.seed(998051)
defenv <- envelope(def, Fest, nrank = 5, nsim = 199)
```

```
## Generating 199 simulations of CSR  ...
## 1, 2, 3, 4.6.8.10.12.14.16.18.20.22.24.26.28.30.32.34.36.38.40
## .42.44.46.48.50.52.54.56.58.60.62.64.66.68.70.72.74.76.78.80
## .82.84.86.88.90.92.94.96.98.100.102.104.106.108.110.112.114.116.118.120
## .122.124.126.128.130.132.134.136.138.140.142.144.146.148.150.152.154.156.158.160
## .162.164.166.168.170.172.174.176.178.180.182.184.186.188.190.192.194.196.198 199.
##
## Done.
```

```
plot(defenv, obs~r, main = 'F plot')
abline(v = 0.06, col = 'red')
```



We observe that at $r = 0.06$ the value of F_{obs} lies below the confidence interval created under the null hypothesis H_0. Hence, we **reject the null hypothesis** that the point pattern is CSR

c)Explain in your own words the differences between the F and G functions. The F and G functions are both similar and in fact have the same value under CSR. However, they differ in that while the G function estimates the distribution of the distance between an event and its neighbor, the F function estimates the distribution of the distance between a random point and the nearest event. As a result, they deviate in opposite directions when the pattern is not CSR.

When points are clustered, distance between an event and its nearest neighbour tends to be small, where as distance between a random point and its nearest event tends to be large. This directly translates into the G function being above the envelope and the F function being below it. The opposite holds when the point pattern is regular.

d) Why is a Monte Carlo test required in this setting?

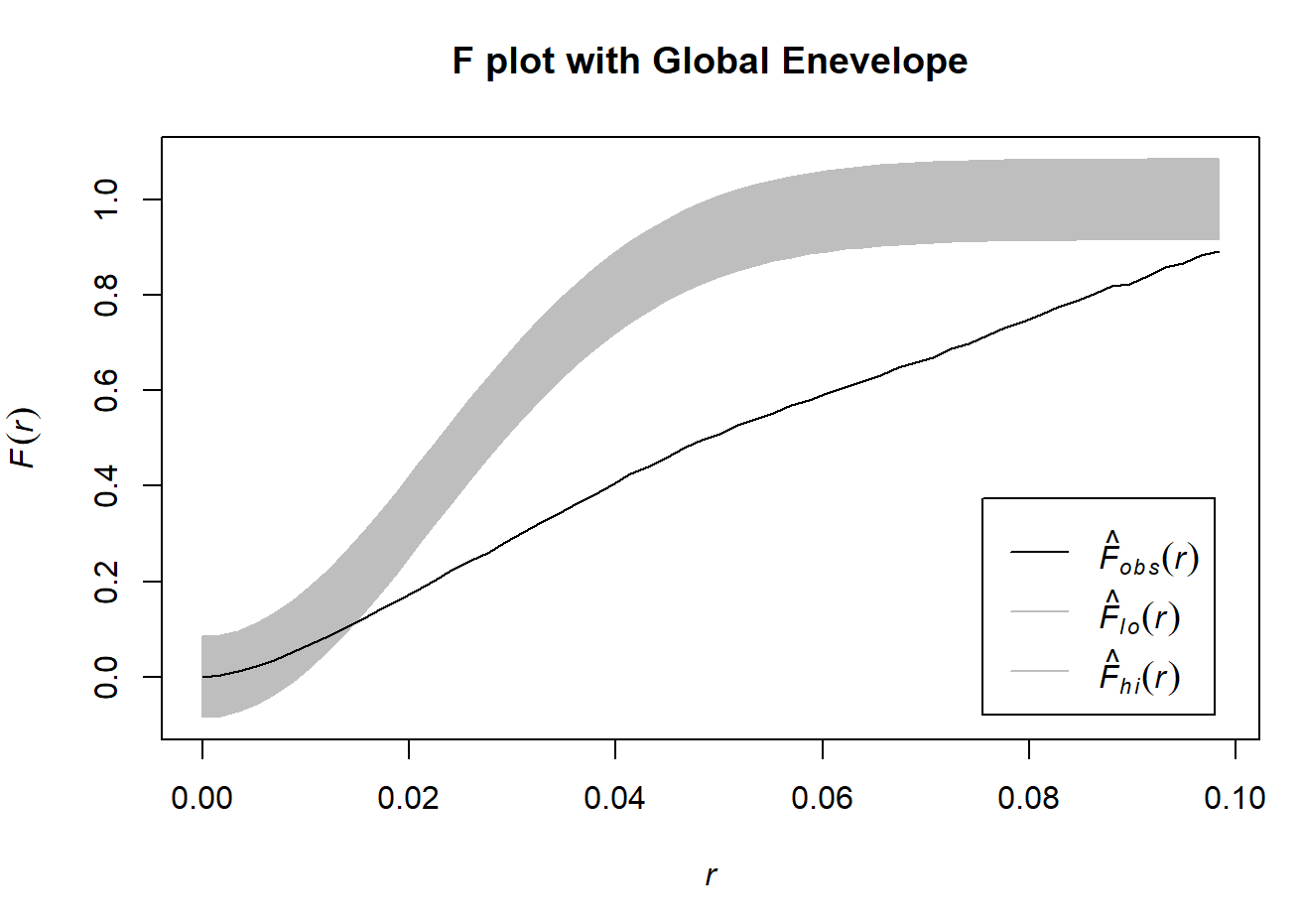
Any analysis of a spatial point pattern is prone to edge effects, which can cause a statistic calculated on the data to be distorted due to points that lie close to the edge of the region under study.This usually results in the theoretical distributions of such statistics being complicated or even unknown. Hence, traditional statistical techniques which cannot capture and account for such distributions due to edge effects can lead to flawed inference. Hence, we resort to Monte Carlo techniques instead by comparing the empirical data under study to simulated datasets under the null hypothesis.

e) Assume we are interested in finding envelopes corresponding to simultaneous confidence bands instead of pointwise confidence intervals. Typing global=TRUE with… function under CSR

```
set.seed(998051)
defenv <- envelope(def, Fest, nrank = 10, nsim = 199, global = TRUE)
```

```
## Generating 199 simulations of CSR  ...
## 1, 2, 3, 4.6.8.10.12.14.16.18.20.22.24.26.28.30.32.34.36.38.40
## .42.44.46.48.50.52.54.56.58.60.62.64.66.68.70.72.74.76.78.80
## .82.84.86.88.90.92.94.96.98.100.102.104.106.108.110.112.114.116.118.120
## .122.124.126.128.130.132.134.136.138.140.142.144.146.148.150.152.154.156.158.160
## .162.164.166.168.170.172.174.176.178.180.182.184.186.188.190.192.194.196.198 199.
##
## Done.
```

```
plot(defenv, obs~r, main = 'F plot with Global Enevelope')
```



Enabling the global option as `True` instructs R to construct the simultaneous envelope rather than a pointwise envelope. Instead of calculating upper and lower bounds for each value of r by using the Monte Carlo realizations of the null hypothesis, simultaneous envelopes involve calculating the maximum deviation of the K function of the observed dataset from each of the simulated datasets under the null hypothesis over all values of `r`. It then chooses the top k_{th} value of the deviation based on the confidence level selected, and uses it to create the upper and lower bounds of the envelope for the K function of the observed dataset. As a result, the width of the envelopeis uniform across all values of `r`

f)Fit a log-linear model of the intensity using covariates x, y, d1 and d2. Write down the model that has been fitted.

```
d1 <- dist$d1
d2 <- dist$d2

model <- ppm(def, ~x + y + d1 + d2)
model
```

```
## Nonstationary Poisson process
##
## Log intensity:  ~x + y + d1 + d2
##
## Fitted trend coefficients:
## (Intercept)           x           y          d1          d2
##   8.0315422   3.1992971  -2.2244723  -0.1673729  -7.8216128
##
##               Estimate      S.E.     CI95.lo    CI95.hi Ztest       Zval
## (Intercept)  8.0315422 0.5627183   6.9286346  9.1344497   *** 14.2727584
## x            3.1992971 1.4793325   0.2998588  6.0987355     *  2.1626627
## y           -2.2244723 0.8059319  -3.8040699 -0.6448747    ** -2.7601242
## d1          -0.1673729 1.5135785  -3.1339323  2.7991866       -0.1105809
## d2          -7.8216128 2.1225468 -11.9817280 -3.6614976   *** -3.6850132
```

The fitted model is

log λ (u) = 8.03 + 3.19 x - 2.22y - 0.16d_1 - 7.82 d_2

g)Perform an ANOVA test, at level /alpha= 0.05, of the model fitted in (f) with a model using only x and y. Compare the AIC of the two models and provide suitable conclusions.

```
model_base <- ppm(def, ~x+y)
anova(model_base, model, test = 'Chi')
```

```
## Analysis of Deviance Table
##
## Model 1: ~x + y   Poisson
## Model 2: ~x + y + d1 + d2     Poisson
##   Npar Df Deviance  Pr(>Chi)
## 1    3
## 2    5  2    43.57 3.458e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(model_base)
```

```
## [1] -1261.892
```

```
AIC(model)
```

```
## [1] -1301.462
```

The p-value of the ANOVA test is significant at the 0.05 level. Additionally, the value of AIC is lower for the second model than for the second one. Hence this suggests that including `d1` and `d2` to estimate the intensity function provides a better model for intensity than the `x` an `y` coordinates alone

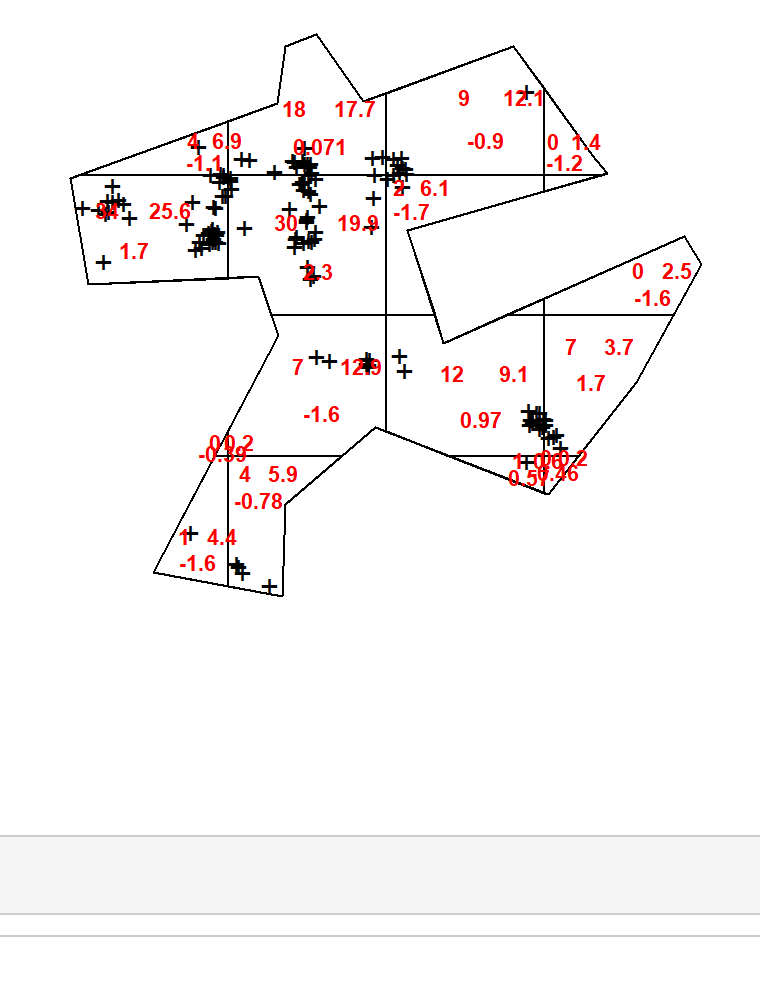
h) Is it true that if models A and B are such that the covariates of B are a subset of A, the AIC of model A is always smaller than that of model B? Why?

This need not always be the case. The way it is calculated, the AIC tries to penalize a model for including extra covariates which might be insignificant. Hence, if the covariates which are present in model A but not in model B are not significant and do not provide any additional information on top of the already existing covariates, then the AIC for model A can be larger than the AIC for model B.However, if the additional covariates provide more information and are significant in trying to estimate a quantity (such as intensity), then the AIC of model A will be smaller than that of model, i.e model A is better than model B.

Additionally, in such a scenario, we can conclude model A to be better than model B only if the difference between the AICs are more than 2 * the number of additionally features in A. This is because AIC adds a difference of 2 for each additional feature by default and hence this needs to be accounted for.

i) Perform a Monte Carlo quadrat test for the log-linear model with covariates x and y, with 4 x 4 quadrats. Intepret the output of the test.

```
set.seed(42380)
quad <- quadrat.test(model_base,nx=4, ny = 4, method = 'MonteCarlo')
plot(def, pch="+", cex = 1)
plot(quad, add=TRUE, col="red", textargs = list(cex = 0.7, font = 2))
```




```
quad
```

```
##
##  Conditional Monte Carlo test of fitted Poisson model 'model_base' using
##  quadrat counts
##  Test statistic: Pearson X2 statistic
##
## data:  data from model_base
## X2 = 27.229, = NA, p-value = 0.096
## alternative hypothesis: two.sided
##
## Quadrats: 16 tiles (irregular windows)
```

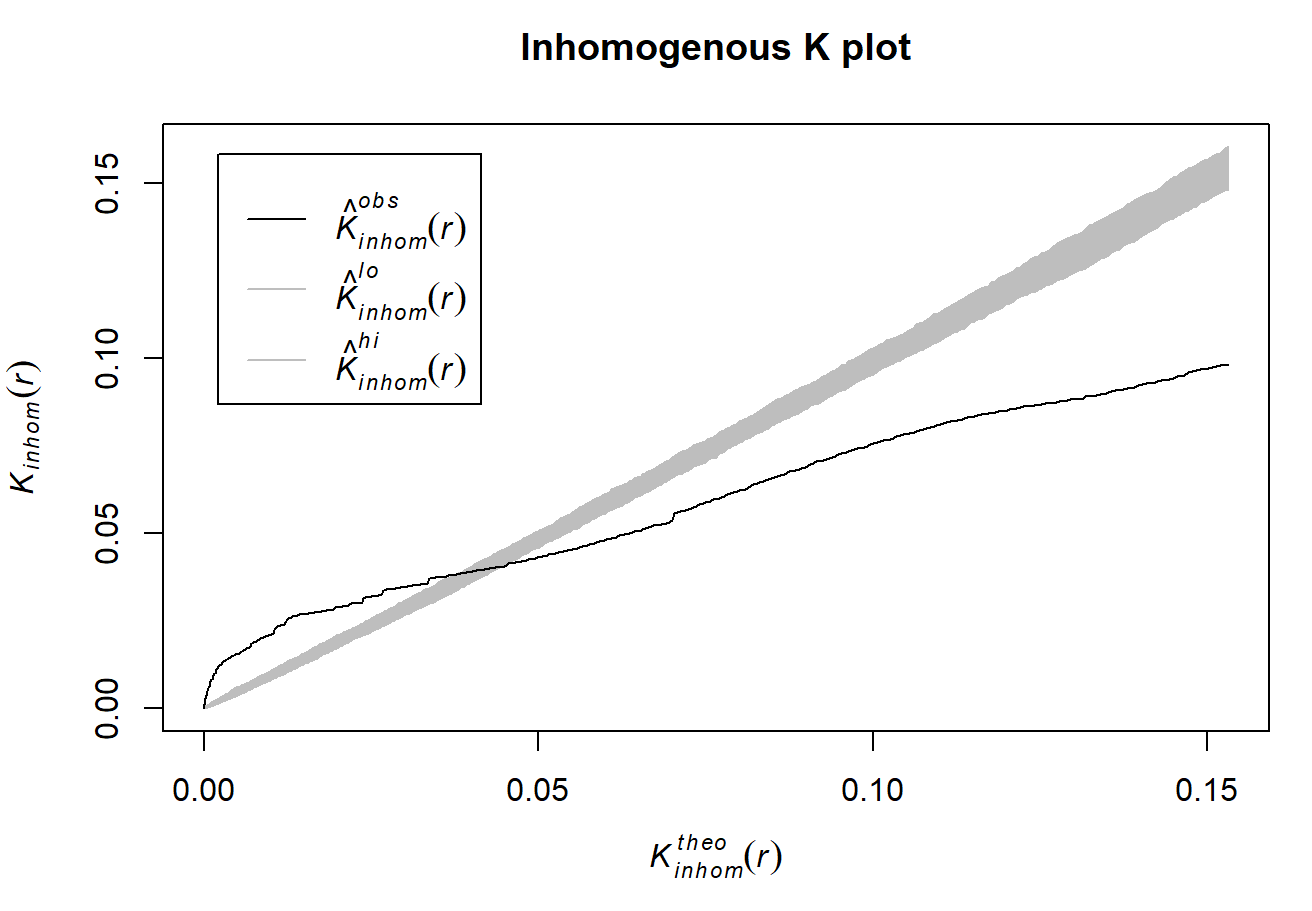
The p-value for the MonteCarlo quadrat test for this model is not significant, and suggesting that the model could be a good fit for the data. We also can not reject IPP as the model dependent on x and y as covariates instead of only a constant seems to be a good fit.

j) The spatstat package allows you to apply the envelope function on Kinhom. You can use the same code as for the F and G functions with def as the ppp object and Kinhom as the function. Plot the Kinhom function of def together with an envelope corresponding to 95% pointwise confidence intervals with 199 simulated datasets.Comment on the plot.

```
set.seed(347)
defenv_kin <- envelope(def, Kinhom, nrank = 5, nsim = 199)
```

```
## Generating 199 simulations of CSR  ...
## 1, 2, 3, 4.6.8.10.12.14.16.18.20.22.24.26.28.30.32.34.36.38.40
## .42.44.46.48.50.52.54.56.58.60.62.64.66.68.70.72.74.76.78.80
## .82.84.86.88.90.92.94.96.98.100.102.104.106.108.110.112.114.116.118.120
## .122.124.126.128.130.132.134.136.138.140.142.144.146.148.150.152.154.156.158.160
## .162.164.166.168.170.172.174.176.178.180.182.184.186.188.190.192.194.196.198 199.
##
## Done.
```

```
plot(defenv_kin, obs~theo, main = 'Inhomogenous K plot')
```



The observed value of the

Inhomogenous K function is first above the envelope for but then crosses over to below the envelope. This suggests that for a small value of r, there is contagion but for larger values, there is competition. An interpretation of this might be that the events occur in clusters, wherein the points within clusters attract each other, however, there are several clusters which are spread out, suggesting the clusters repel each other.

3. a The number of events N in the region A follows a Poisson distribution with mean

$$C_\lambda = \int_A \lambda(u)\, du$$

$$\therefore P(N = n) = \frac{e^{-\int_A \lambda(u) du} \left(\int_A \lambda(u) du\right)^n}{n!}$$

where, $\log(\lambda(u)) = \beta_0 + \sum_{j=1}^{p} \beta_j z_j(u)$

b It is given that density

$$f(u) \propto \lambda(u)$$

$$\Rightarrow f(u) = k\,\lambda(u), \text{ k is some constant}$$

But $\int_A f(u)\, du = 1$

$$\therefore k \int_A \lambda(u)\, du = 1$$

$$\Rightarrow k = \frac{1}{\int_A \lambda(u)\, du} = \frac{1}{C_\lambda}$$

$$\therefore f(u) = \frac{\lambda(u)}{C_\lambda}$$

c Let

$$L^*(\beta) = P(N = n) \prod_{i=1}^{n} f(u_i)$$

$$= \frac{e^{-C_\lambda} (C_\lambda)^n}{n!} \prod_{i=1}^{n} \frac{\lambda(u_i)}{C_\lambda} \quad \left[\text{From a \& b}\right]$$

$$= \frac{e^{-C_\lambda} (C_\lambda)^n}{n!} \times \frac{1}{(C_\lambda)^n} \prod_{i=1}^{n} \lambda(u_i)$$

$$\therefore \log(L^*(\beta)) = -C_\lambda$$

$$= \frac{e^{-C_\lambda}}{n!} \times \prod_{i=1}^{n} \lambda(u_i)$$

$$\therefore \log(L^*(\beta)) = -C_\lambda + \log\left[\prod_{i=1}^{n} \lambda(u_i)\right] - \log(n!)$$

$$= \sum_{i=1}^{n} \log \lambda(u_i) - \int_A \lambda(u)\, du - \log(n!)$$

$$\therefore \log(L^*(\beta)) = L(\beta) - \log(n!)$$

or

$$L(\beta) = \log(L^*(\beta)) - \log(n!)$$

Hence, $L(\beta)$ is an increasing function of $L^*(\beta)$. As a result of this, the β which maximizes $L^*(\beta)$ & the β which maximizes $L(\beta)$ are the same.

Question 4

a) Describe bei and bei.extra with the help of the class and str commands.

```
data_description <- list(bei_class = class(bei),
                         bei_struct = capture.output(str(bei)),
                         bei_ext_class = class(bei.extra),
                         bei_ext_str = capture.output(str(bei.extra)))

for (item in 1:4)
    print(data_description[item])
```

```
## $bei_class
## [1] "ppp"
##
## $bei_struct
##  [1] "List of 5"
##  [2] " $ window    :List of 4"
##  [3] "  ..$ type  : chr \"rectangle\""
##  [4] "  ..$ xrange: num [1:2] 0 1000"
##  [5] "  ..$ yrange: num [1:2] 0 500"
##  [6] "  ..$ units :List of 3"
##  [7] "  .. ..$ singular  : chr \"metre\""
##  [8] "  .. ..$ plural    : chr \"metres\""
##  [9] "  .. ..$ multiplier: num 1"
## [10] "  .. ..- attr(*, \"class\")= chr \"unitname\""
## [11] "  ..- attr(*, \"class\")= chr \"owin\""
## [12] " $ n         : int 3604"
## [13] " $ x         : num [1:3604] 11.7 998.9 980.1 986.5 944.1 ..."
## [14] " $ y         : num [1:3604] 151 430 434 426 415 ..."
## [15] " $ markformat: chr \"none\""
## [16] " - attr(*, \"class\")= chr \"ppp\""
##
## $bei_ext_class
## [1] "imlist"  "solist"  "anylist" "listof"  "list"
##
## $bei_ext_str
##  [1] "List of 2"
##  [2] " $ elev:List of 10"
##  [3] "  ..$ v     : num [1:101, 1:201] 121 121 121 121 121 ..."
##  [4] "  ..$ dim   : int [1:2] 101 201"
##  [5] "  ..$ xrange: num [1:2] -2.5 1002.5"
##  [6] "  ..$ yrange: num [1:2] -2.5 502.5"
##  [7] "  ..$ xstep : num 5"
##  [8] "  ..$ ystep : num 5"
##  [9] "  ..$ xcol  : num [1:201] 0 5 10 15 20 25 30 35 40 45 ..."
## [10] "  ..$ yrow  : num [1:101] 0 5 10 15 20 25 30 35 40 45 ..."
## [11] "  ..$ type  : chr \"real\""
## [12] "  ..$ units :List of 3"
## [13] "  .. ..$ singular  : chr \"metre\""
## [14] "  .. ..$ plural    : chr \"metres\""
## [15] "  .. ..$ multiplier: num 1"
## [16] "  .. ..- attr(*, \"class\")= chr \"unitname\""
## [17] "  ..- attr(*, \"class\")= chr \"im\""
## [18] " $ grad:List of 10"
## [19] "  ..$ v     : num [1:101, 1:201] 0.252 0.202 0.162 0.17 0.207 ..."
## [20] "  ..$ dim   : int [1:2] 101 201"
## [21] "  ..$ xrange: num [1:2] -2.5 1002.5"
## [22] "  ..$ yrange: num [1:2] -2.5 502.5"
## [23] "  ..$ xstep : num 5"
## [24] "  ..$ ystep : num 5"
## [25] "  ..$ xcol  : num [1:201] 0 5 10 15 20 25 30 35 40 45 ..."
## [26] "  ..$ yrow  : num [1:101] 0 5 10 15 20 25 30 35 40 45 ..."
## [27] "  ..$ type  : chr \"real\""
## [28] "  ..$ units :List of 3"
## [29] "  .. ..$ singular  : chr \"metre\""
## [30] "  .. ..$ plural    : chr \"metres\""
## [31] "  .. ..$ multiplier: num 1"
## [32] "  .. ..- attr(*, \"class\")= chr \"unitname\""
## [33] "  ..- attr(*, \"class\")= chr \"im\""
## [34] " - attr(*, \"class\")= chr [1:5] \"imlist\" \"solist\" \"anylist\" \"listof\" ..."
```

b) The as function is a general function that lets you convert R objects… the gradient (slope) in bei.extra to Spatial-GridDataFrame.

```
bei_sp <- as(bei, "SpatialPoints")
bei_extra_slope <- as(bei.extra$grad, "SpatialGridDataFrame")
```

c) Create a SpatialPointsDataFrame with the tree locations in bei as the point pattern and the slope at the tree locations as the covariates.Display the first few entries of the SpatialPointsDataFrame.

```
slope <- over(bei_sp, bei_extra_slope)
colnames(slope) <- c('slope')

bei_slope <- SpatialPointsDataFrame(coordinates(bei_sp), slope)

head(bei_slope)
```

```
##      coordinates     slope
## 1  (11.7, 151.1) 0.1161989
## 2 (998.9, 430.5) 0.2846527
## 3 (980.1, 433.5) 0.2227576
## 4 (986.5, 425.8) 0.2417234
## 5 (944.1, 415.1) 0.0852658
## 6 (940.5, 410.4) 0.0914678
```