DATA589_Project

Justin Chan, CM Tong (Kenny)

2023-04-14

1. Introduction

1.1 Ascomycota as a phylum of the kingdom Fungi

Ascomycota is a phylum (or category) of the kingdom Fungi. In fact, it is the largest phylum of Fungi. Its members are also commonly known as sac fungi.

Among many other Fungi phylum, Ascomycota is of particularly useful to humans. As it can be used as sources of medicinally important compounds (e.g. antibiotics), and fermentation for various food products such as alcoholic beverages, bread and cheese. One famous example of its members is Penicillium, which has been widely used and utilized in natural environment, in food spoilage, and food and drug production.

The Ascomycota also plays an important role in most land-based ecosystems, because they are efficient decomposers, breaking down organic materials (e.g. dead leaves and animals) and completing the nutrient cycle.

1.2 Data Source

Data is downloaded from GBIF with the download link below : URL: https://www.gbif.org/occurrence/download/0165170-230224095556074

1.3 Research Questions

Due to the potential medically and various practical values of Ascomycota, we would be interested in exploring and investigating its growth distribution, patterns and population density in terms of spatial analysis.

During this project, we hope to answer the following questions :

- Are there any spatial patterns?
- Is the distribution homogeneous and inhomogeneous?
- Are there any significant correlation with the available covariate data such as Elevation, Forest, and Distance to Water?
- Are there evidence to support any occurrence clustering, independence or avoidance? Are these affected by the homogeneity and inhomogeneity assumption of the data?
- What are the possible prediction models? Does complicated model outperform simple model? Is the complicated model worth it?
- Can higher-degree polynomial model further improve the prediction model?

1.4 References

- https://www.britannica.com/science/fungus/Outline-of-classification-of-fungi
- https://en.wikipedia.org/wiki/Penicillium
- https://en.wikipedia.org/wiki/Ascomycota

2. Methods and Results

2.1 Dataset Variables and Selection

A total of about 69,000 occurrences records are available in BC, which are consolidated from a total of 109 datasets. However, the following three major parties constitute about 85% of the total occurrences records :

- UBC Herbarium - Licen Collection
- iNaturalist Research-grade Observations
- Assembly and activity of microbial communities in the Pacific temperate rainforest

```
# install.packages("rgbif")
library(rgbif)
#?rgbif

species <- c("Ascomycota")

# Get number of occurrence records from rgbif
#occ_count(scientificName = species) #16069587 in the dataset

#Filter Ascomycota data to only in BC
asc_count <- occ_count(scientificName = species,
                          hasCoordinate = TRUE,
                          country = "CA",
                          stateProvince = "British Columbia")

asc_bc_data <- occ_data(scientificName = species,
                           hasCoordinate = TRUE,
                           country = "CA",
                           stateProvince = "British Columbia",
                           limit=2000) #* 102 illegal points stored in attr(,"rejects") ***

# class(asc_bc_data) #gbif_data
asc_bc_data <- asc_bc_data$data #gbif_data to data.frame
## head(asc_bc_data) #contain "Ascomycota" data only in BC
```

2.1.1 Data Records

Firstly, screen through some sample records of the dataset to see what information it contains and which attributes would and would not be useful and valuable in our analysis.

A total of 75 data colummns are available. These columns can be broadly divided into the following three main categories : - Scientific Classification and Taxonomy Details : Scientific Name, Hierarchy of Biological classification and taxonomic ranks, etc - Point and Location Details : Continent, State/Province, Coordinates, Coordinates Uncertainty Meters, etc - Data Collection Details : Collector, Date and Timestamp

2.1.2 Data Cleaning

As the dataset contains too many information, including many detailed timestamps, internal key/identifiers information and dataset/records identifiers which should not be valuable in our analysis and complicate our subsequent analysis, we have performed preliminary cleaning procedure and performed attributes selection, in order to only retain those few potentially useful attributes, speeding up our subsequent analysis and making it more focused. The cleaned list iis shown below.

```r
# View(data.frame(names(asc_bc_data)))
cleaned_asc_bc <- asc_bc_data[ , c("decimalLongitude", "decimalLatitude", "order", "family", "genus", "s
                                   "year", "month", "day", "eventDate", "occurrenceStatus", "class", "co
                                   "verbatimEventDate", "collectionCode", "gbifID", "verbatimLocality"
                                   )]
```

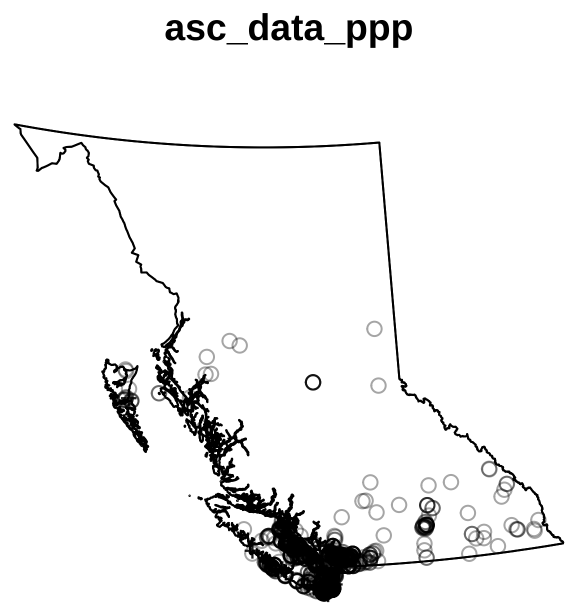
```r
names(cleaned_asc_bc)
```

```
##  [1] "decimalLongitude"              "decimalLatitude"
##  [3] "order"                         "family"
##  [5] "genus"                         "species"
##  [7] "genericName"                   "specificEpithet"
##  [9] "coordinateUncertaintyInMeters" "stateProvince"
## [11] "year"                          "month"
## [13] "day"                           "eventDate"
## [15] "occurrenceStatus"              "class"
## [17] "countryCode"                   "country"
## [19] "verbatimLocality"              "taxonID"
## [21] "catalogNumber"                 "institutionCode"
## [23] "eventTime"                     "verbatimEventDate"
## [25] "collectionCode"                "gbifID"
## [27] "verbatimLocality.1"
```

2.1.3 Spatial Dataset Object Conversion and Preparation

As the current format of the fungi data is not in a PPP class object, to facilitate subsequent analysis and plots, we would firstly convert it to the PPP, which would make subsequent plotting and analysis library much more accessible.

```r
plot(asc_data_ppp)
```



2.2 Statistical Packages

The following packages are used to perform the analysis : - rgbif : allow searching and retrieving data from GBIF, access various dataset metadata, species names, and occurences details - sp, rgdal : process, manipulate and transform the data source longitude and latitude information - spstat : provides many statistical tests, analysis and plots - maptools : provides utilities to handle and analyze spatial objects - Others supporting libraries include kdensity and splines

2.3 Analytical Workflow

The report will start with some basic Exploratory Data Analysis and basic data plottings to have an initial understanding of the overall data points pattern.

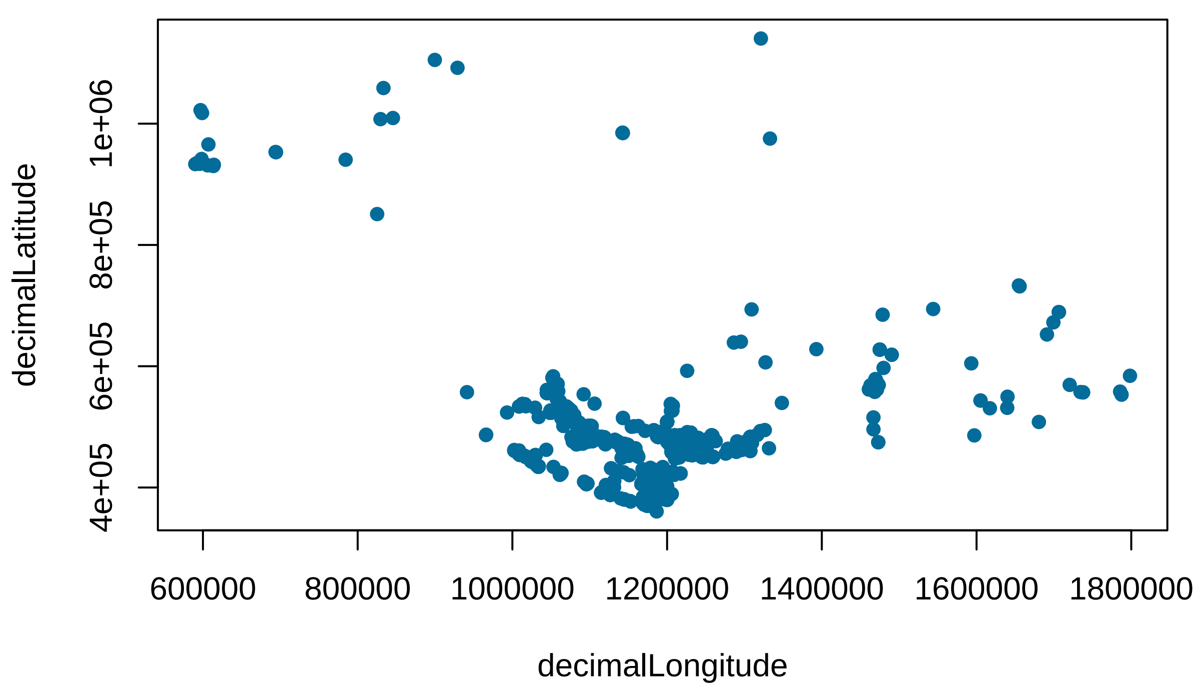
We will then conduct First Moment Analysis which will include homogeneity analysis and covariates study on some variables. Afterwards, we will perform Second Moment analysis where popular tools such as K-function and Pair Correlation Function will be deployed to investigate any clustering.

The report will then conduct model fitting and selection with both lower and higher polynomial variable terms and evaluate the fitness and its costs with AIC values. The report is concluded by a final model validation process for the resulted model to assess the wellness of the model fitting.

2.3.1 EDA : Initial Coordinates Plotting

First, we have a coordinate plot to observe the general pattern in terms of the coordinates.

```
#Visualise the data
plot(decimalLatitude ~ decimalLongitude,
     pch = 16,
     col = "#046C9A",
     data = cleaned_asc_bc)
```



From the initial coordinate plot, we have identified the following very preliminary observation :

- Points clustering is highly likely as a large group of data points is spotted at the decimal Longitude between 1000K and 1300K and Latitude between 4e+5 and 5e+5.

- Therefore is only a single major and significant large clustering only. Although there are some other scattered points observed near the Longitude range 1500K-1800K, they are simply incomparable to the large clustering.

2.3.2 EDA : BC Windows and Covariates Data Walkthough

Let's see what the BC_Covariates.Rda file provide us which may help us to identify and choose appropriate fields to be used in the covariates analysis in the later part of this analysis. It should contain the information on the BC province :

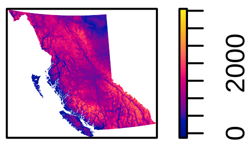
- Windows
- Elevation
- Forest
- Dist Water

```
#plot(BC_win, pch = 16, cols = "red", main = "BC windows data")
```

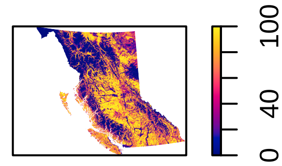
```
par(mfrow=c(2,2))

#Elevation
plot(BC$Elevation, main = "Elevation")
#Forest
plot(BC$Forest, main = "Forest")
#Dist_Water
plot(BC$Dist_Water, main = "Distance to Water")
plot(asc_data_ppp,
     which.marks = "species", # Which mark to use
     col = "grey", #The colour of the window
     cols = 'red', #The colours of the points
     cex = 0.6,
     pch = 18, # The plotting symbol
     main = "Ascomycota in BC", # The title
     par(bg="grey40", cex.main = 2),
     cex = 0.6,
     legend = T) # Turn off the legend depending on needs
```

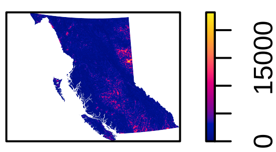
Elevation



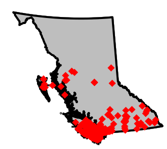
Forest



Distance to Water



Ascomycota in BC



Next, We try to pick Elevation to divide its values into 5 levels, and see if there are any patterns for the distribution of the fungi points in the different elevation level.

```
cut <- cut(BC$Elevation,5,
labels = c("low","low-medium","medium","medium-high","high"))
table(cut[asc_data_ppp]) #most in low elevation
```

```
##
##         low  low-medium      medium medium-high        high
##        1481          67           8           0           0
```

Finding : - Overwhelming majority of the fungi points are found in the low elevation region

2.3.3 KDE Analysis

```
# nn_dist <- nndist(asc_data_ppp)
# marks(asc_data_ppp) <- nn_dist
# plot(asc_data_ppp, main = "Ascomycota Distance",which.marks = "Dist", pch = 16)
BC$Elevation[asc_data_ppp]
```

```
##    [1]   91.2716797   91.2716797   23.3979000   62.6924058   51.4287909
##    [6]  261.9802821  261.9802821  261.9802821   74.4980033   74.4980033
##   [11]  510.9264437   32.1182483   62.8145572   81.5482938   81.5482938
```

```
##  [16]   76.0844928  706.4096761   65.1946191    8.0000000  -35.1796687
##  [21]  -35.1796687  -35.1796687  -35.1796687   76.0844928   76.0844928
##  [26]   76.0844928   32.1182483   32.1182483  195.2845866   61.6651847
##  [31]  206.5882499   66.0000000   65.6194786  288.0706032  288.0706032
##  [36]  288.0706032  288.0706032  288.0706032  355.9678917  355.9678917
##  [41]  355.9678917  355.9678917  355.9678917  355.9678917  355.9678917
##  [46]  288.0706032  288.0706032  288.0706032  288.0706032  288.0706032
##  [51]  288.0706032  288.0706032  288.0706032  288.0706032  288.0706032
##  [56]  288.0706032  430.1152729  430.1152729  288.0706032  288.0706032
##  [61]  355.9678917  430.1152729  430.1152729  288.0706032  288.0706032
##  [66]  288.0706032  288.0706032  288.0706032  288.0706032  225.4816113
##  [71]  225.4816113  198.1439260   81.1268103  -35.0538969  -35.0538969
##  [76]   39.3694513   39.3694513  294.3840245  -35.0538969  -35.0538969
##  [81]  -35.0538969  -35.0538969  -35.0538969  -35.0538969   71.8017864
##  [86]  -35.0538969  187.9576960  187.9576960  171.9366133  167.1073329
##  [91]   40.3963313   40.3963313   40.3963313   40.3963313   40.3963313
##  [96]   40.3963313 1038.4348810  777.0987040 1670.9647397   69.2897891
## [101]   94.4226083   69.2897891   39.0920756   82.5834767   14.6985124
## [106]   55.8831146   36.3969244   49.1268926   45.6529487   50.7014826
## [111]   50.7014826   50.7848676   49.1268926   44.2464938   25.4281845
## [116]   25.4281845   25.4281845  339.5916067   21.0000000   21.0000000
## [121]   21.0000000   25.1411527 1119.1858518  108.9600329   45.6529487
## [126]   32.1182483   23.8995600   92.1437063   42.8808548   54.0691653
## [131]   50.7848676  130.7351456  147.0059584   50.7848676 1318.6351898
## [136]  131.2367039  114.9395210  131.2367039  160.0908808   44.2464938
## [141] 1130.7751076   46.4328392   -3.7349688   20.7143934   44.0000000
## [146]  122.4896565   53.9855684  177.9525474   46.4328392  117.7496930
## [151]   50.7848676   46.4328392   72.7176954  433.1949152   33.9959414
## [156]   72.7176954   72.7176954   48.8967005   25.5473702   55.2437213
## [161]   55.2437213    2.8956890   32.1182483  197.6772909   46.4401374
## [166]   46.4401374   71.8017864  198.1439260   23.3979000   32.1182483
## [171]   80.8255768  102.0710382  433.7867630  224.3301012  224.3301012
## [176]  189.2972383  214.4397175  189.9900552  203.3261835  203.3261835
## [181]  203.3261835  203.3261835  203.3261835  206.5882499   22.0340832
## [186]   18.2429452  -35.0538969   45.6573633  510.9264437  510.9264437
## [191]  197.6772909  197.6772909  -57.4582138   60.8336975   50.7848676
## [196]  206.5882499  -35.0538969  -35.0538969   32.1182483  181.3066385
## [201]   27.6813483   27.6813483  176.4374541  176.4374541   37.0000000
## [206]   32.1182483   46.4328392   26.5203128   62.8145572    4.4120537
## [211]  315.8078534  315.8078534   40.6094565  135.8061649   50.7848676
## [216]   66.7418292   31.2127656   53.5079504  267.1413641  233.3740888
## [221]  129.4073228  197.6772909 1119.1858518   67.9014289  145.8768265
## [226]  235.7033300  334.2387537 1108.3876856  206.4887597  206.4887597
## [231]  206.4887597  193.4001865  115.0197268   32.1182483   27.4074114
## [236]   32.1182483   32.1182483   32.1182483   32.1182483   44.2464938
## [241]   44.2464938   26.5203128   26.5203128   26.5203128   26.5203128
## [246]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [251]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [256]   26.5203128   26.5203128  404.8587849   26.5203128   26.5203128
## [261]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [266]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [271]   26.5203128   26.5203128   32.1182483   32.1182483   43.8526123
## [276]   43.8526123   43.8526123   26.5203128   26.5203128   26.5203128
## [281]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
```

```
## [286]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [291]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [296]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [301]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [306]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [311]   62.8145572   62.8145572   62.8145572   62.8145572   62.8145572
## [316]   64.1561569   23.8041106   45.6529487   26.5203128   36.3969244
## [321]   -6.2446304  123.1643173  205.3635116  205.3635116  205.3635116
## [326]  118.9491512  118.9491512  118.9491512  138.1870224   68.1811070
## [331]   45.6529487   45.6529487   45.6529487   45.6529487   50.7848676
## [336]   50.7848676   45.6529487   27.6813483  115.0197268  115.0197268
## [341]  115.0197268   32.1182483   32.1182483 1555.4483105    9.1073183
## [346]   45.6529487   18.0139750   26.3520622   26.5203128   26.5203128
## [351]   26.5203128   26.5203128   26.5203128   66.7418292   66.7418292
## [356]   66.7418292   27.6813483   45.6529487  -35.0538969   26.5203128
## [361]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [366]   26.5203128   26.5203128   26.5203128   26.5203128   26.5203128
## [371]   26.5203128   38.4799246   20.0000000   22.1192646   20.3160487
## [376]   20.3160487  767.5566690   34.8474945  172.7646671   79.0501967
## [381]  231.5900532  253.8222738  201.5487542   55.9460750  147.2731380
## [386]   10.0000000   42.5046879   61.6651847  168.8637969  117.9323929
## [391]   32.1182483   32.1182483   46.4328392  113.9998772  113.9998772
## [396]   20.4014117   13.7439203 1161.1355816  180.6210158  180.6210158
## [401]    5.0000000   85.7916291   85.7916291   85.7916291   85.7916291
## [406]   85.7916291   85.7916291   85.7916291   85.7916291   99.8309870
## [411]   99.8309870  102.0710382  102.0710382   85.7916291   85.7916291
## [416]   27.6490344   99.8309870   99.8309870   99.8309870   99.8309870
## [421]   99.8309870   99.8309870   99.8309870   99.8309870   99.8309870
## [426]   99.8309870   99.8309870   99.8309870   99.8309870   99.8309870
## [431]   99.8309870   99.8309870   99.8309870   99.8309870  114.9395210
## [436]   31.7535827  268.2274521   40.3963313  268.2274521  267.3107792
## [441]   29.5669432   29.5669432   29.5669432   45.6529487   63.3549512
## [446]   78.2745755   45.6529487   20.4014117   20.4014117    1.0000000
## [451]  110.1031573 1760.3323586   45.6529487 1353.6952866   45.6529487
## [456]   23.8995600   80.2378827   45.6529487   28.8456018   28.8456018
## [461]   28.8456018   18.7505410   32.1182483   53.2566740   44.2464938
## [466]   46.4328392   92.7112108   92.7112108   13.7439203   27.2552979
## [471]   27.2552979   56.7136338   45.6529487  581.9027372 1269.1941151
## [476]  206.5882499   48.2921024  825.4552251   46.4328392  462.5532324
## [481]  350.9813115  206.5882499   70.4358509   22.1192646   40.3963313
## [486]   40.3963313   40.3963313  179.3045843   46.4328392   46.4328392
## [491]   46.4328392   46.4328392   46.4328392   46.4328392   46.4328392
## [496]   18.3345062   18.3345062  128.5077361   32.7914995   22.8287039
## [501]    8.0000000    8.0000000    8.0000000    8.0000000    8.0000000
## [506]    8.0000000    8.0000000    8.0000000    8.0000000    8.0000000
## [511]    8.0000000    7.2121578   20.9835775  128.5077361 1313.3230973
## [516]   20.4014117  128.5077361    8.0000000    8.0000000    8.0000000
## [521]    8.0000000    8.0000000    8.0000000   45.6529487   45.6529487
## [526]    8.0000000   14.0000000    8.0000000   17.2237405   45.6529487
## [531]   45.6529487   45.6529487   45.6529487   44.2464938  158.2756167
## [536]   58.0000000   58.0000000  195.2845866    6.0851279  360.4243019
## [541]  360.4243019  360.4243019  360.4243019  248.4005832  248.4005832
## [546]  105.3055229  248.4005832  275.8426970  275.8426970  275.8426970
## [551]  248.4005832  248.4005832  248.4005832  144.7747909  144.7747909
```

```
## [556]   86.4564476  144.7747909   45.6529487   18.2429452   18.2429452
## [561]   18.2429452   85.7916291   85.7916291   85.7916291   85.7916291
## [566]   85.7916291   85.7916291   57.3615949   57.3615949   57.3615949
## [571]   32.1182483   32.1182483   65.4236637   65.4236637   19.9989742
## [576]   64.1366998   61.0000000  510.9264437   62.7797407   99.1959306
## [581]   99.1959306   99.1959306  101.4595918  101.4595918  101.4595918
## [586]  120.9216017  120.9216017   22.0340832   21.0000000  115.0197268
## [591]   -7.4597171   20.4014117   11.4103130   31.3371049   26.9630346
## [596]   27.2952824   81.2522907   81.2522907  195.2845866   31.3371049
## [601]   31.3371049   31.3371049   91.1593248   31.3371049   31.3371049
## [606]   31.3371049   62.8145572   45.1544140   45.1544140   45.1544140
## [611]   80.6439755  112.8723323   38.3263899  112.6830370  112.6830370
## [616]   61.6651847   81.5367242    7.2121578   44.2464938   71.8017864
## [621]  355.9678917   44.2464938   63.7425153   22.5291811   22.5291811
## [626]   22.5291811   22.5291811   81.5482938   63.7425153    5.8102222
## [631]   18.5839850   18.5839850   18.5839850   63.7425153   19.7216533
## [636]   28.0214855   28.0214855   28.1432034   28.1988832   28.1988832
## [641]   28.1988832   28.1988832   28.1988832   28.1988832   28.1988832
## [646]  355.9678917   28.1988832   28.1988832   28.1988832   28.1988832
## [651]   28.1988832   40.3963313   18.3345062   19.4150918   19.4150918
## [656]   19.4150918    5.8102222    5.8102222    5.8102222   39.4847608
## [661]   34.8474945   34.8474945   39.4847608   28.1988832   28.1988832
## [666]   28.1988832   28.1988832   28.1988832   28.1988832   28.1988832
## [671]   28.1988832   28.1988832   79.0992980   13.8600001 1178.5118635
## [676]  139.0401548   65.1946191   45.6529487   28.5627558  112.1472987
## [681]   44.9479602   10.0000000   76.0844928  206.5882499 1139.2157061
## [686]   27.2552979   76.0844928   36.3969244   27.2314184  117.7496930
## [691]   19.6143969   71.4688522   64.1366998   64.1366998   84.6775035
## [696]  -35.3836371   32.1182483   32.1182483  -35.3836371  -35.3836371
## [701]   26.8976393   22.1192646   22.1192646   21.1242654   21.1242654
## [706]   21.1242654   22.1192646  117.9323929   36.3969244  755.8722562
## [711]  755.8722562  755.8722562  755.8722562   23.3979000   26.8976393
## [716]   29.5798158  339.5916067  323.3657927  323.3657927  248.1315986
## [721]  248.1315986  248.1315986  141.4680988  141.4680988  141.4680988
## [726]  128.5077361  100.0460062  128.5077361  624.0648656  128.5077361
## [731]  128.5077361  128.5077361   37.9146424   37.9146424   37.9146424
## [736]   36.9118647   36.9118647   22.9130695   64.8387920   64.8387920
## [741]   64.8387920   59.1940794   59.1940794   59.1940794   59.1940794
## [746]  660.7383622  660.7383622  186.8805546  318.7594806   20.4014117
## [751]   20.4014117 1160.1078032  541.9698024   81.5367242  114.6205783
## [756]   61.7246401   46.0222595   46.0222595  310.9811526  310.9811526
## [761]   -2.0000000   19.1026843   19.5805807   19.5805807   19.5805807
## [766]   19.5805807   19.5805807   19.5805807   19.1026843   19.1026843
## [771]   19.1026843   22.6064859   22.6064859    8.0000000  145.2480183
## [776]  300.9232303   36.3969244   12.0000000   36.3393342   19.4150918
## [781]   13.5915868   13.5915868   13.5915868   13.5915868   13.5915868
## [786]    7.6394210    7.6394210    7.6394210    7.6394210    7.6394210
## [791]    7.6394210    7.6394210    8.0719579 1623.0571110 1623.0571110
## [796]   44.2464938   45.6529487   42.0263913   64.8387920   62.8145572
## [801]   50.3024259  248.1315986  248.1315986   32.9855163   44.2464938
## [806]   17.6862495  342.0588191  455.6587562   86.2121445  235.1647589
## [811]   66.9231863   66.9231863   66.9231863   66.9231863   66.9231863
## [816]   66.9231863   66.9231863   85.7177928   85.7177928   85.7177928
## [821]   85.7177928   85.7177928   85.7177928   85.7177928   85.7177928
```

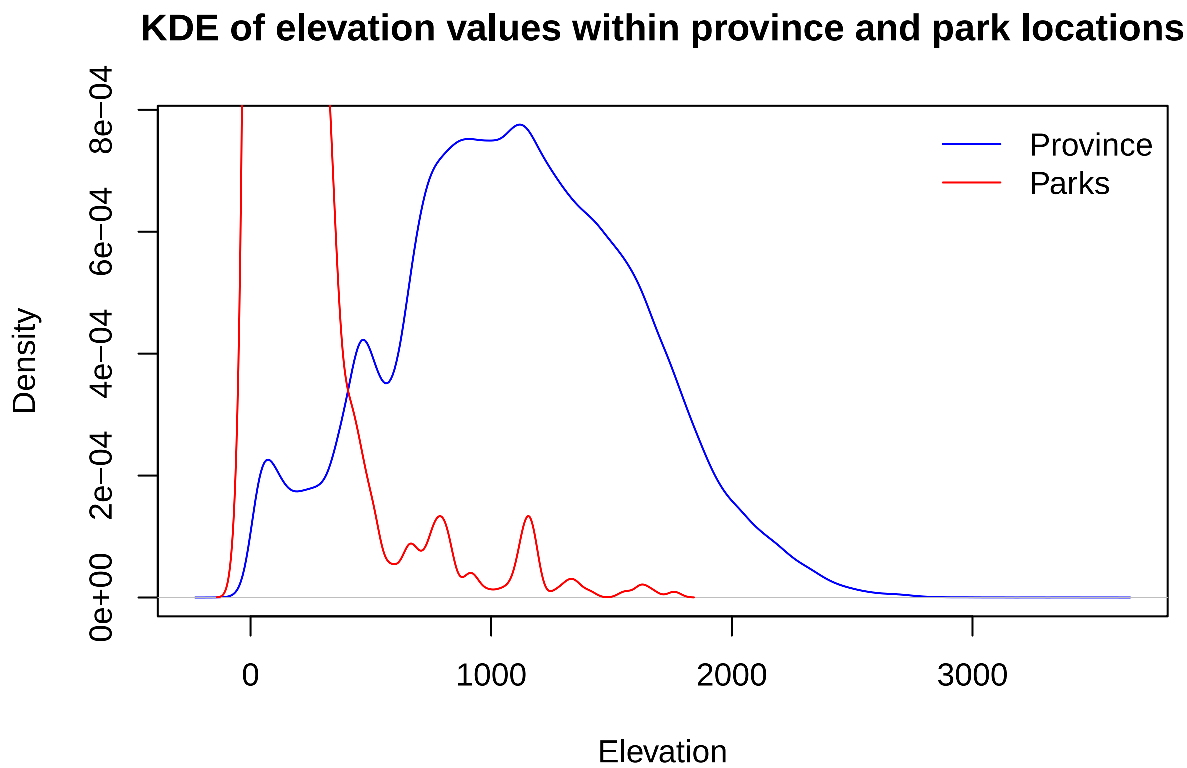
```
##  [826]    85.7177928   69.2013097  749.7214785  125.9785533    8.9535612
##  [831]   433.0043899    2.8956890   27.2952824  197.6413892   81.5367242
##  [836]    79.5775173   70.2446672   57.3615949  424.2770036  267.3107792
##  [841]   267.3107792  267.3107792  267.3107792   44.2464938   74.4980033
##  [846]    74.4980033   74.4980033   74.4980033   74.4980033   74.4980033
##  [851]    74.4980033   74.4980033   74.4980033   74.4980033   36.3969244
##  [856]   592.3631396   78.7934925   78.7934925  407.8466491   70.5775792
##  [861]    70.5775792   70.5775792   70.5775792   70.5775792   70.5775792
##  [866]    70.5775792   32.1182483  133.7293092   32.1182483   39.0920756
##  [871]    26.5203128  120.9216017    8.0000000  127.4526369  127.4526369
##  [876]   120.9216017  127.4526369  127.4526369  127.4526369  127.4526369
##  [881]   302.0297425  302.0297425  302.0297425   65.4236637   86.7249572
##  [886]    82.5834767   82.5834767   82.5834767  -35.3836371  -35.3836371
##  [891]    42.1613053  102.0710382  375.2219779  375.2219779  131.2367039
##  [896]   178.3273465  178.3273465  131.2367039  131.2367039  113.4815429
##  [901]   336.7795638  336.7795638  336.6993326  382.9967441  385.7130922
##  [906]   385.7130922  336.7795638   23.3979000   45.6529487  213.6968575
##  [911]   106.2430334  336.7795638  336.7795638  336.7795638  336.7795638
##  [916]   336.7795638  336.7795638  181.0013082  163.6414619  381.5455956
##  [921]   328.4422446  823.3279871  142.9677749  134.8170391   10.0000000
##  [926]    10.0000000   45.6529487   64.1561569  518.4621195   19.6143969
##  [931]  1156.5999095 1178.5118635 1178.5118635  247.6463257  247.6463257
##  [936]   247.6463257   57.1989549    0.5410833   27.2314184  248.1315986
##  [941]   248.1315986  248.1315986  248.1315986  248.1315986  248.1315986
##  [946]   323.3657927  323.3657927  323.3657927  323.3657927  323.3657927
##  [951]   323.3657927  323.3657927  323.3657927  323.3657927  323.3657927
##  [956]   323.3657927  323.3657927  323.3657927  323.3657927  323.3657927
##  [961]   323.3657927  323.3657927  323.3657927  323.3657927  323.3657927
##  [966]   120.5178817  823.3279871  195.2845866   45.6529487   -2.3717241
##  [971]     0.5850159   44.2464938  177.9525474  247.7763524  286.1536123
##  [976]   177.9525474  195.2845866  177.9525474  177.9525474  177.9525474
##  [981]   177.9525474   13.7439203   13.7439203   13.7439203   61.7288302
##  [986]    88.1842872   74.0020248   13.7439203  227.2961897  302.0297425
##  [991]    54.1990245  846.5764224   55.8831146   19.2687107   19.2687107
##  [996]    19.2687107   50.7848676  206.5882499  195.2845866   45.6529487
## [1001]   302.0297425  302.0297425  302.0297425  302.0297425    5.7833656
## [1006]    44.2464938   62.8145572   85.7916291  113.9998772  267.3107792
## [1011]    25.0197592  211.1629177  525.0080937  913.3582545  127.9098280
## [1016]   127.2527046   76.1974346  224.5557096  577.8621472   78.7934925
## [1021]    32.1182483   15.4046650   -2.3717241   74.5352022   81.2325555
## [1026]    23.3979000   20.4014117   46.0222595    8.0000000    8.0000000
## [1031]    65.1946191   65.1946191   57.3615949   57.3615949   62.9234208
## [1036]    90.6371252   25.4281845   79.0992980   55.8831146  231.5900532
## [1041]   115.0197268  253.7300640  253.7300640  289.2261889  289.2261889
## [1046]   289.2261889  253.7300640  253.7300640   45.6529487  231.5900532
## [1051]    46.4401374   22.9495438   66.5280620   74.2874187  289.2261889
## [1056]    33.9889193   45.6529487   45.6529487   71.4688522   55.2437213
## [1061]    25.4248680   25.4248680   48.9552566  181.2001848   14.6985124
## [1066]    14.6985124   14.6985124   14.6985124   63.3549512   27.0000000
## [1071]    60.6152609   60.6152609   60.6152609  358.0274712  899.0700159
## [1076]   209.6782699  209.6782699  209.6782699  209.6782699  209.6782699
## [1081]   209.6782699  209.6782699   64.1561569  358.0274712  358.0274712
## [1086]   358.0274712  209.6782699   23.3979000   23.3979000   23.3979000
## [1091]    23.3979000   23.3979000   23.3979000   49.0218679   70.7616098
```

```
## [1096]    56.3793654   56.3793654   56.3793654  109.3899973   46.4328392
## [1101]    78.5379588   65.6194786   61.6651847   61.6651847   61.6651847
## [1106]    61.6651847   61.6651847  221.3087334  899.0700159   50.7014826
## [1111]    50.7014826   54.0691653  395.0737739  398.9876029   32.1182483
## [1116]   219.7955638  187.6207135  187.6207135  219.7955638    5.0000000
## [1121]   322.3069738  293.1005047   16.2427151    5.0000000   32.1182483
## [1126]    18.0139750   18.0139750   45.6529487  140.9568825  140.9568825
## [1131]    39.5917382   30.0122775   18.0139750   18.0139750   18.0139750
## [1136]    18.0139750   18.0139750   18.0139750   18.0139750  213.6968575
## [1141]     3.1594155    3.1594155    3.1594155    3.1594155    6.8647719
## [1146]     6.8647719    6.8647719   36.3969244   32.0314691   73.2209453
## [1151]    24.1074388  101.3083959   97.8916388   84.7104652  405.3397510
## [1156]    67.3983017  691.5729341  170.8297441   74.1568270  170.8297441
## [1161]    53.5965499   53.5965499   53.5965499   53.5965499   53.5965499
## [1166]    20.7047846   42.0263913   42.1733608   56.6232019   44.2464938
## [1171]   121.9823043   65.4236637   24.1074388   24.1074388   26.2287883
## [1176]   253.7300640  253.7300640  289.2261889  350.0416609  659.5173410
## [1181]   659.5173410  659.5173410  659.5173410  637.6135893   -6.2446304
## [1186]    -6.2446304   -6.2446304   -6.2446304  214.4397175  214.4397175
## [1191]   303.8685924  351.8048279  987.1065742  282.2060671  282.2060671
## [1196]   282.2060671  282.2060671  282.2060671  282.2060671  282.2060671
## [1201]    61.6651847   61.6651847   61.6651847   61.6651847  253.8222738
## [1206]   231.5900532   61.6651847   62.8145572  178.6086350   11.1562874
## [1211]   178.6086350  303.8685924  303.8685924  303.8685924  303.8685924
## [1216]   303.8685924  303.8685924  303.8685924  303.8685924   24.0000000
## [1221]    45.3003284   33.9889193   33.9889193   33.9889193   10.0000000
## [1226]   178.6086350  178.6086350  122.4896565  160.0908808   46.4401374
## [1231]    70.4358509   70.4358509   52.0202352   70.4358509  228.2791278
## [1236]   115.0197268  115.0197268  154.0839301  404.8587849  404.8587849
## [1241]   404.8587849  298.5627336  298.5627336  404.8587849  404.8587849
## [1246]   404.8587849  298.5627336   36.0272506   85.7034675  183.7834237
## [1251]  1148.2725438 1148.2725438 1148.2725438   70.1771578   88.1842872
## [1256]    45.6529487   32.1182483  822.6584252  147.4981123   75.8806177
## [1261]    23.3979000   74.0020248   69.2013097   74.0020248  218.7441402
## [1266]   218.7441402  218.7441402  218.7441402  218.7441402  218.7441402
## [1271]   218.7441402   69.2013097   74.2874187  250.0369742  404.8587849
## [1276]   500.9837290   64.1561569   44.2464938  934.6215645 1075.1735530
## [1281]    64.1561569  420.7826041  376.4665121  404.8587849  404.8587849
## [1286]    22.8295629   22.8295629  343.1633106   71.7304594   75.5499662
## [1291]   294.2864411  294.2864411  443.3989115  443.3989115  443.3989115
## [1296]   443.3989115  477.5133425  477.5133425  477.5133425  443.3989115
## [1301]   219.7888331  219.7888331  346.0011839  206.5882499   71.7304594
## [1306]    45.6529487   45.6529487   26.9630346   32.9233925  140.2657815
## [1311]   140.2657815  700.2108487  934.6215645 1133.3143096  829.3412007
## [1316]    40.3963313  721.9678195   27.6813483  133.7403911   20.1626588
## [1321]    78.7934925   75.8806177   75.8806177   75.8806177   13.7439203
## [1326]    45.6529487   45.6529487  187.9576960  187.9576960  187.9576960
## [1331]   187.9576960  137.4361789   45.6529487   50.7848676   50.7848676
## [1336]    23.3979000   14.6985124   14.6985124   18.0139750   18.0139750
## [1341]    14.6985124   14.6985124   14.6985124   14.6985124   14.6985124
## [1346]    14.6985124   14.6985124   14.6985124  106.0904007   35.9033676
## [1351]    35.9033676   35.9033676  253.7300640  465.6242671  465.6242671
## [1356]   527.1842418  527.1842418  221.3087334   75.8806177  760.5142727
## [1361]    77.1493716  163.6414619  163.6414619  218.8889551  218.8889551
```

```
## [1366]   77.1493716  100.4141159  156.6989651  156.6989651   79.3706117
## [1371]  218.8889551  218.8889551  218.8889551   99.4124315   76.9815890
## [1376]   77.1493716   77.1493716   76.9815890  112.1305149  253.7256670
## [1381]   72.1776463  156.6989651   76.9815890   76.9815890   76.9815890
## [1386]   76.9815890   76.9815890   90.3978849   76.9815890   45.6529487
## [1391]   14.6985124   14.6985124   14.6985124   14.6985124  156.6989651
## [1396]  131.3760501  131.3760501  131.3760501  131.3760501  131.3760501
## [1401]  131.3760501  290.5139005  109.3899973  100.4141159  100.4141159
## [1406]   13.7439203  290.5139005  278.0593169  278.0593169  278.0593169
## [1411]   74.4980033   74.0020248  156.6989651  156.6989651  131.3760501
## [1416]   45.6529487   45.6529487  131.3760501  131.3760501  131.3760501
## [1421]  156.6989651  156.6989651  156.6989651  131.3760501  156.6989651
## [1426]  156.6989651  131.3760501  156.6989651  131.3760501  156.6989651
## [1431]  156.6989651  156.6989651  131.3760501  156.6989651  156.6989651
## [1436]   14.6985124  156.6989651  156.6989651  156.6989651  156.6989651
## [1441]  218.8889551  218.8889551  455.6587562  478.5888668  478.5888668
## [1446]  478.5888668 1178.5118635 1178.5118635   76.9815890   76.9815890
## [1451]   10.2591957   23.1101247   23.1101247   23.1101247 1178.5118635
## [1456]   45.2030910   -8.3685426  129.4073228   76.0516586  478.5888668
## [1461]  154.2785337   32.1182483   40.3963313  500.1776076  167.1073329
## [1466]  167.1073329  154.2785337   52.7587042  164.3433635  164.3433635
## [1471]  164.3433635  154.2785337  154.2785337  154.2785337  154.2785337
## [1476]   37.0000000  118.7287075   77.1677506  798.0119248  798.0119248
## [1481]  798.0119248  798.0119248  798.0119248  798.0119248  253.7300640
## [1486]  253.7300640  253.7300640  253.7300640  253.7300640  253.7300640
## [1491]  253.7300640  253.7300640 1408.3834663  167.1073329 1351.5718803
## [1496]   11.4981224  794.2982699  720.0539972  253.7300640  253.7300640
## [1501]  253.7300640  253.7300640  253.7300640  253.7300640  253.7300640
## [1506]  568.1925510   34.5610641   34.5610641  108.8702820   29.4038398
## [1511]  424.2770036  424.2770036   54.0691653   19.7216533   52.1288732
## [1516]  164.3433635 1146.1984392  511.5474308   46.5549923  146.9407390
## [1521]  146.9407390  146.9407390  146.9407390  146.9407390  146.9407390
## [1526]  146.9407390  146.9407390  104.3013978  146.9407390  146.9407390
## [1531]  767.3473086  462.5532324   91.2716797   26.9630346   91.2716797
## [1536]   66.7418292   27.2952824   56.3793654   56.3793654   56.3793654
## [1541]   56.3793654   56.3793654   79.4646302  224.9294300   13.8600001
## [1546]  193.4452303  382.9967441  382.9967441  456.3934640  682.2886794
## [1551]  102.7541076  102.7541076   56.3793654   62.6170421  608.0374890
## [1556]  154.3547530
```

```
library("kdensity")
asc_density <- density(BC$Elevation[asc_data_ppp])
province_density <- density(BC$Elevation$v, na.rm=T)


plot(province_density, main = "KDE of elevation values within province and park locations", xlab = "Ele
lines(asc_density, col = "red")
legend("topright", legend = c("Province", "Parks"), lty = 1, col = c("blue", "red"), bty = "n")
```




```
#
# plot(province_density, main = "KDE of elevation values within province and park locations", xlab = "E
# lines(park_density, col = "red")
# legend("topright", legend = c("Province", "Parks"), lty = 1, col = c("blue", "red"), bty = "n")
```

2.3.4 First Moment Descriptive Statistics

After preliminary EDA and high level plot, we will study various first moment descriptive statistical measures.

```
intensity(asc_data_ppp)
```

2.3.5.a Intensity

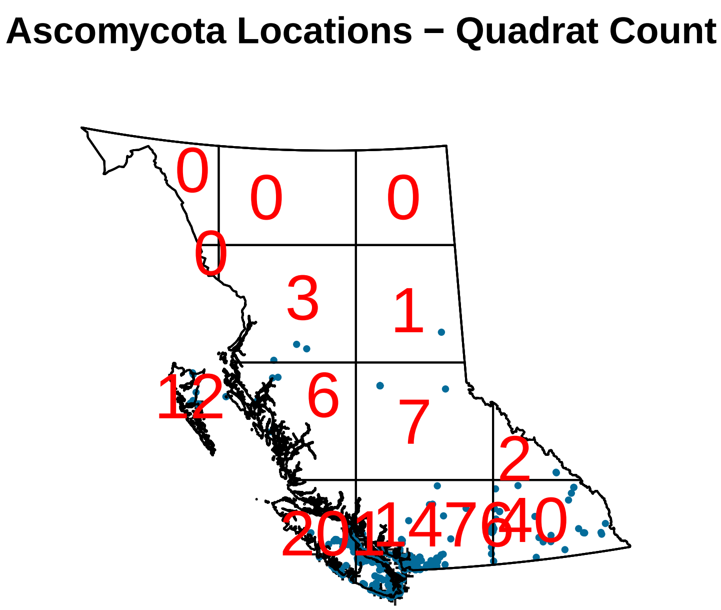
```
## [1] 1.843372e-09
```

2.3.5.b Homogeneity Studies, Quadrat Test and Hotspot Analysis Note that the intensity is a very small number. This is consistent with our plots above. There are not many points in the whole BC windows. Overwhelming majority of the points are spresely located in the BC. Let's verify this with the Quadrat Count Plot.

```r
Q <- quadratcount(asc_data_ppp,
                  nx = 4,
                  ny = 4)

#Plot the output
plot(asc_data_ppp,
     pch = 16,
     cex = 0.5,
     cols = "#046C9A",
     main = " Ascomycota Locations - Quadrat Count")

plot(Q, cex = 2, col = "red", add = T)
```

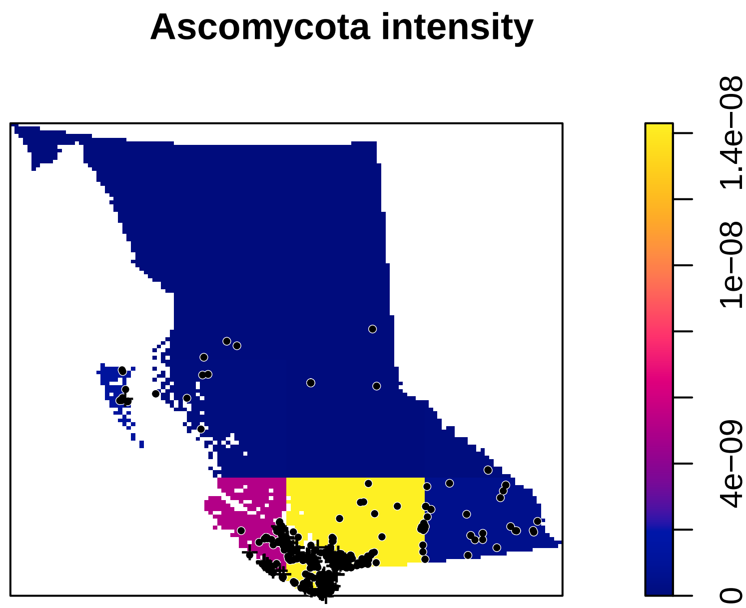



```r
plot(intensity(Q, image = T),
     main = "Ascomycota intensity")

plot(asc_data_ppp,
     pch = 16,
     cex = 0.6,
     cols = "white",
     add = T)

plot(asc_data_ppp,
     pch = 16,
     cex = 0.5,
```

```
    cols = "black",
    add = T)
```



Next, we perform a Qudart test of homogeneity

```
#Quadrat test of homogeneity
quadrat.test(Q)
```

```
##
##  Chi-squared test of CSR using quadrat counts
##
## data:
## X2 = 10279, df = 12, p-value < 2.2e-16
## alternative hypothesis: two.sided
##
## Quadrats: 13 tiles (irregular windows)
```

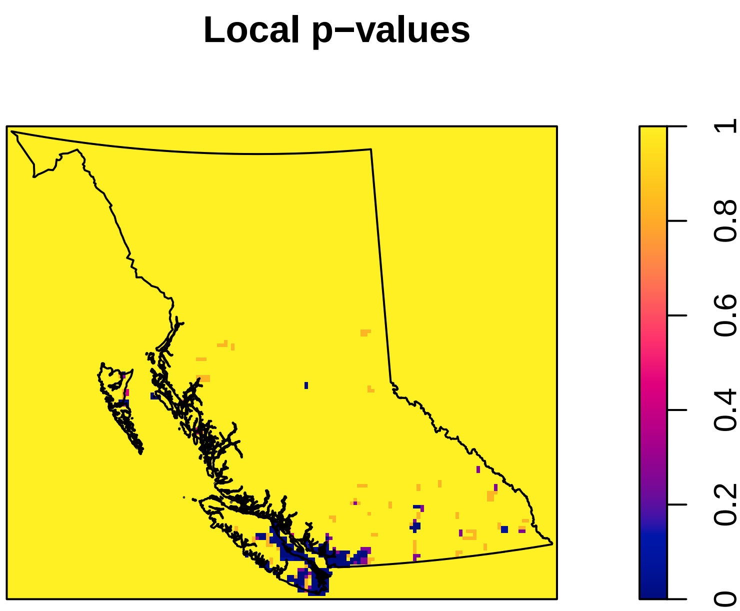
In addition, We also perform a Likelihood Ratio Test to evaluate the degree of homogeneity as cross reference.

```
R <- bw.ppl(asc_data_ppp)
LR <- scanLRTS(asc_data_ppp,r=R)

#plot(LR,"Likelihood Ratio Test")
#plot(asc_data_ppp$window,"Likelihood Ratio Test")
```

```r
pvals <- eval.im(pchisq(LR,
                         df = 1,
                         lower.tail = FALSE))


#Plot the output
plot(pvals, main = "Local p-values")
plot(asc_data_ppp$window,add=T)
```



Once again, these plots have strongly verified the spatial inhomogeneity nature of the fungi distribution :

- Overwhelming majority of the regions have zero or near zero points.
- Quadrat Test and Likelihood Ratio Test indicates a significant deviation from homogeneity for the points.
- Hot spot analysis shows only a very few prominent hot spot with low p-values. Most regions are purely yelliw (i.e. p=1).

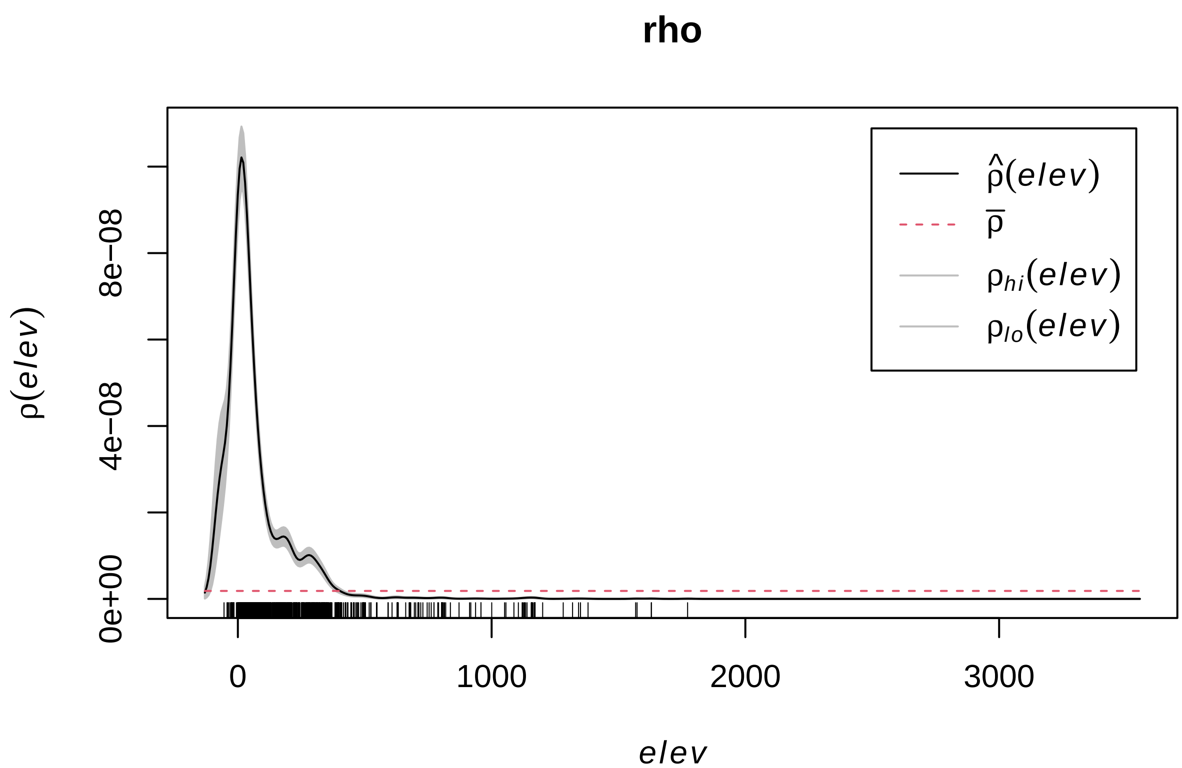
2.3.6 Covariate Study

Now, we will study the covariate behaviour individually, and see whether there are strong support or observable patterns. We will conduct quantile split into 4 sections for each of the covariates.

2.3.6.a Covariate Variable : Elevation Let's start with Elevation.

```
elev <- BC$Elevation
b <- quantile(elev,probs=(0:4)/4,type=2)
Zcut <- cut(elev,breaks=b)
V <- tess(image=Zcut)
quadratcount(asc_data_ppp,tess=V)
```

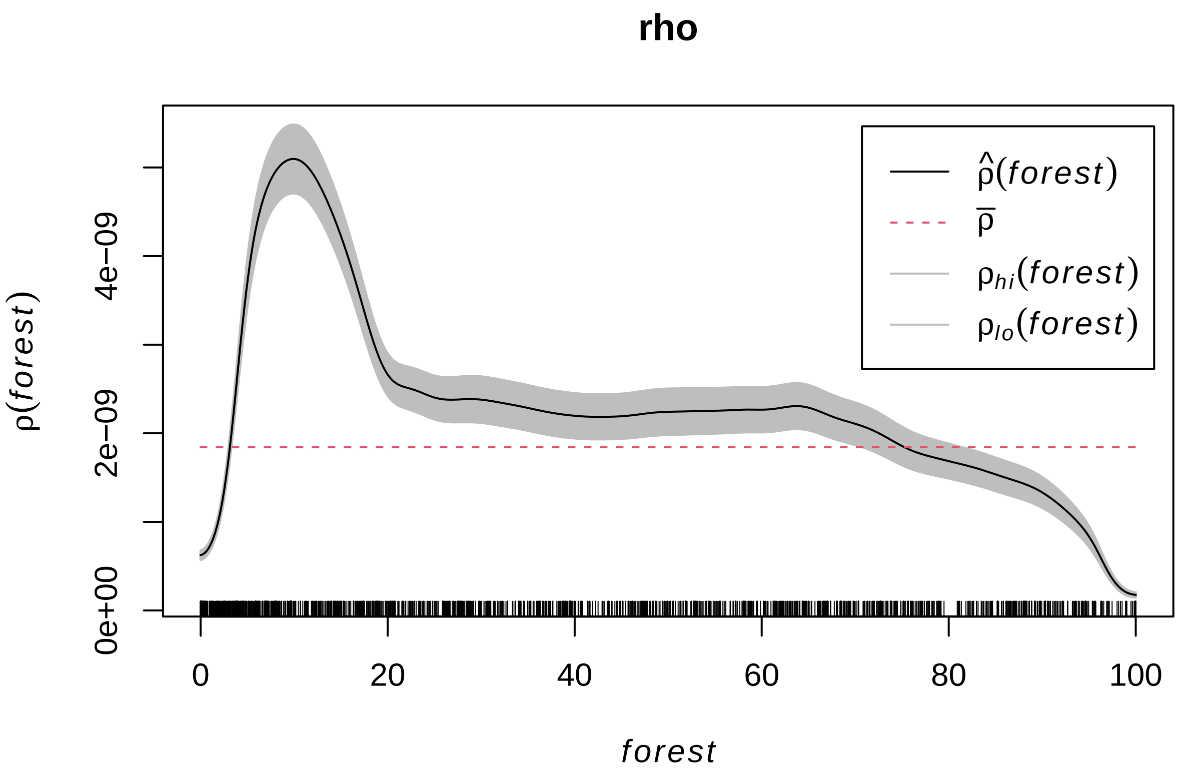
```
## tile
##          (-130,761]       (761,1.1e+03]   (1.1e+03,1.46e+03]  (1.46e+03,3.56e+03]
##                1694                  24                   25                    5
```



2.3.6.b Covariate Variable : Forest Then, let's see Forest.

```
forest <- BC$Forest
b <- quantile(forest,probs=(0:4)/4,type=2)
Zcut <- cut(forest,breaks=b)
V <- tess(image=Zcut)
quadratcount(asc_data_ppp,tess=V)
```

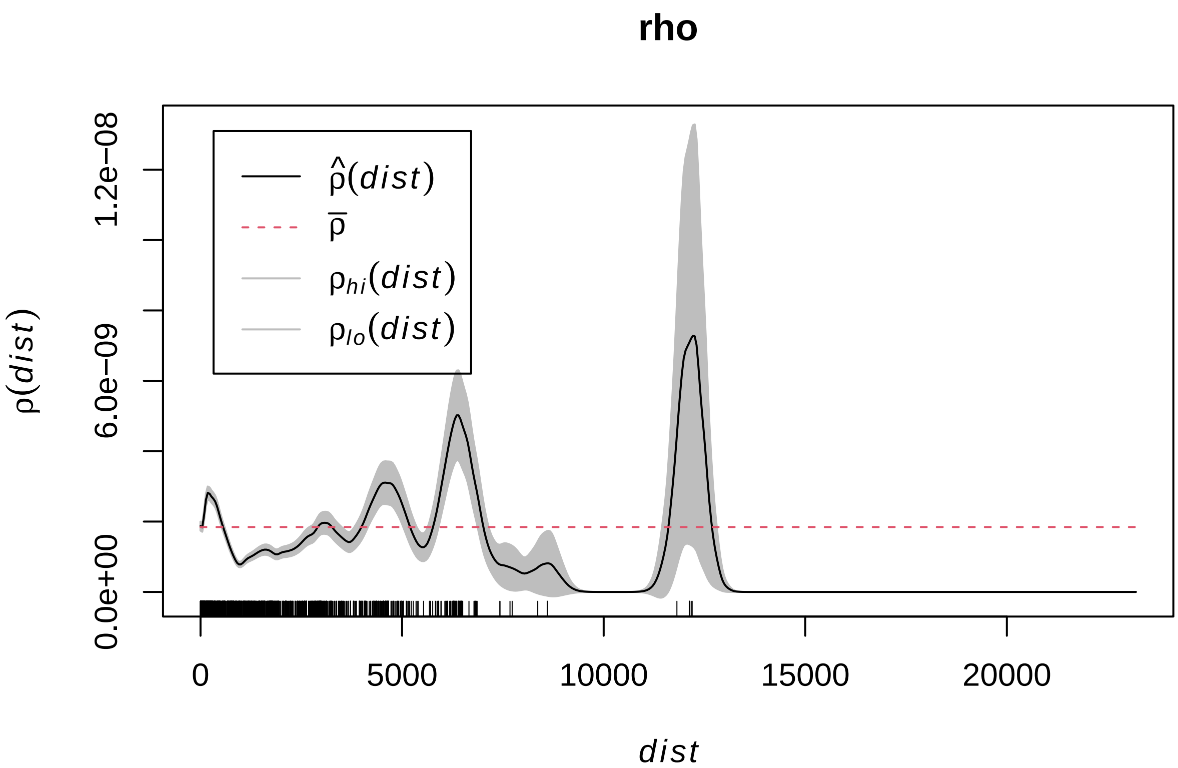
```
## tile
##    (0,11.6] (11.6,50.2] (50.2,86.9]  (86.9,100]
##         582         533         455         177
```



2.3.6.c Covariate Variable : Distance to Water Followed by Distance to Water.

```
dist <- BC$Dist_Water
b <- quantile(dist,probs=(0:4)/4,type=2)
Zcut <- cut(dist,breaks=b)
V <- tess(image=Zcut)
quadratcount(asc_data_ppp,tess=V)
```

```
## tile
##            (0,483]        (483,1.1e+03]  (1.1e+03,2.18e+03] (2.18e+03,2.32e+04]
##                785                  220                  267                  424
```



2.3.6.d Section Conclusion Observation :

- For Elevation Level :
  - It is obvious that the fungi is overwhelmingly correlated to the low elevation level.
  - The proportion of occurrence that appears in the lowest elevation sector accounts for more than 97% of the identified points.
- For Forest and Distance to Water :
  - It is observed that the fungi is correlated to both Forest and distance .
  - The proportion of occurrence that appears in the respective smallest sectors account for about 50% of the identified points.

2.3.7 Second Moment Descriptives

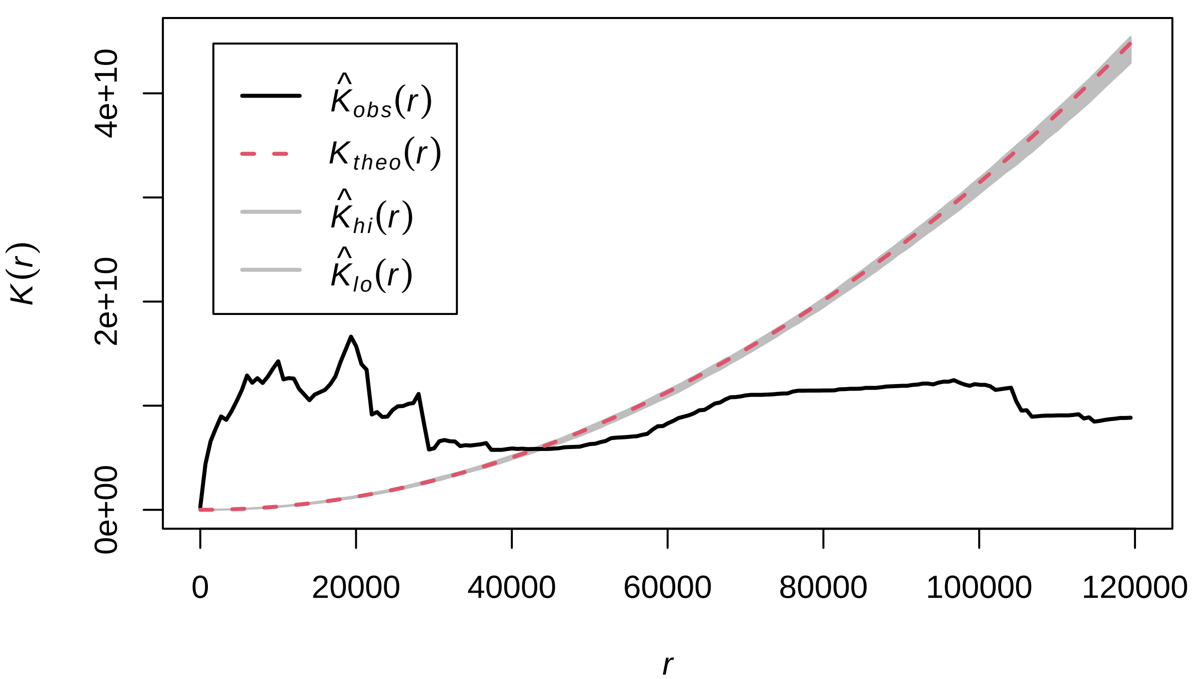
To continue the analysis, we will explore the second moment descriptives to uncover any variances and correlation characteristics of the data.

2.3.7.a Ripley's K-function We firstly use the typical K-function to measure the spatial clustering and point pattern.

```r
# Bootstrapped CIs
# rank = 1 means the max and min
# Border correction is to correct for edges around the window
# values will be used for CI
E_asc <- envelope(asc_data_ppp,
                  Kest,
                  correction="border",
                  rank = 1,
                  nsim = 19,
                  fix.n = T)
```

```
## Generating 19 simulations of CSR with fixed number of points  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,  19.
##
## Done.
```

```r
# visualise the results
plot(E_asc,
     main = "",
     lwd = 2,
     xlim=c(0,120000))
```



As inhomogeneity has been observed, we are going to correct for inhomogeneity and reperform the test for a more accurate result.

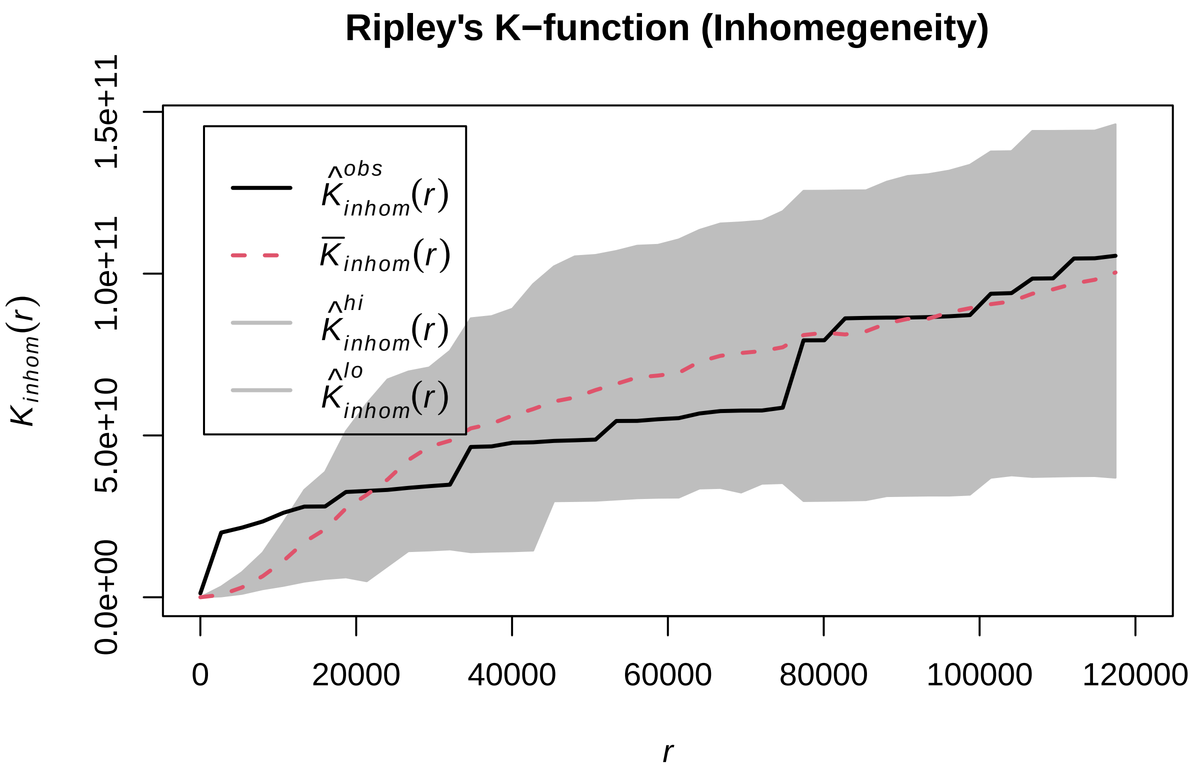
```r
lambda_asc <- density(asc_data_ppp, bw.ppl)
Kinhom_asc <- Kinhom(asc_data_ppp, lambda_asc)

#Estimate a strictly positive density
lambda_asc_pos <- density(asc_data_ppp,
                          sigma=bw.ppl,
                          positive=TRUE)

#Simulation envelope (with points drawn from the estimated intensity)
E_asc_inhom <- envelope(asc_data_ppp,
                        Kinhom,
                        simulate = expression(rpoispp(lambda_asc_pos)),
                        correction="border",
                        rank = 1,
                        nsim = 19,
                        fix.n = TRUE)
```

```
## Generating 19 simulations by evaluating expression  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,  19.
##
## Done.
```

```r
# visualise the results
# par(mfrow = c(1,2))
plot(E_asc_inhom,
     main = "Ripley's K-function (Inhomegeneity)",
     lwd = 2,
     xlim=c(0,120000))
```



Finding :

- When assumed homogeneity, the observed data show a strong clustering at small r value range.
- When corrected for inhomogeneity, such clustering pattern is no longer observed.

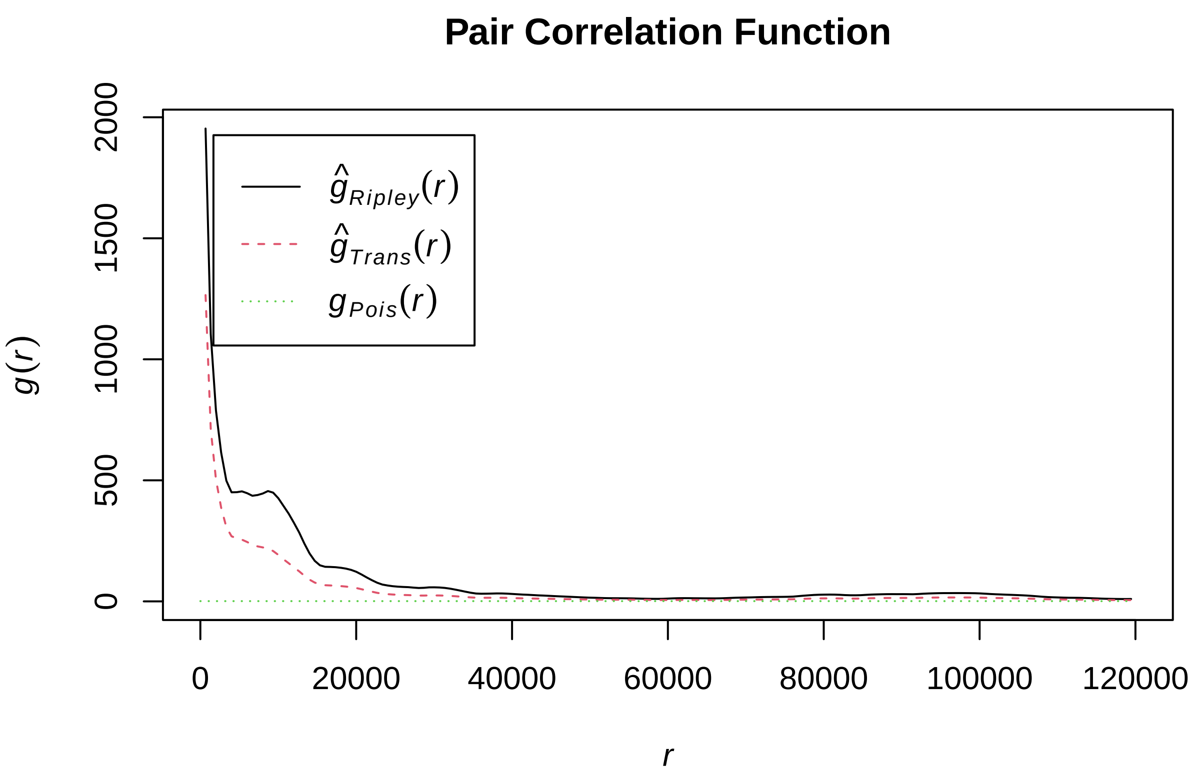
2.3.7.b Pair Correlation Function We are also interested in checking the points relation using pair correlation function.

```
# Estimate the g function
pcf_asc <- pcf(asc_data_ppp)

pcf_asc
```

```
## Function value object (class 'fv')
## for the function r -> g(r)
## .................................................................
##       Math.label          Description
## r     r                   distance argument r
## theo  g[Pois](r)          theoretical Poisson g(r)
## trans hat(g)[Trans](r)    translation-corrected estimate of g(r)
## iso   hat(g)[Ripley](r)   isotropic-corrected estimate of g(r)
## .................................................................
## Default plot formula:  .~r
## where "." stands for 'iso', 'trans', 'theo'
## Recommended range of argument r: [0, 341660]
## Available range of argument r: [0, 341660]
```

```r
plot(pcf_asc,xlim=c(0,120000),main="Pair Correlation Function")
```

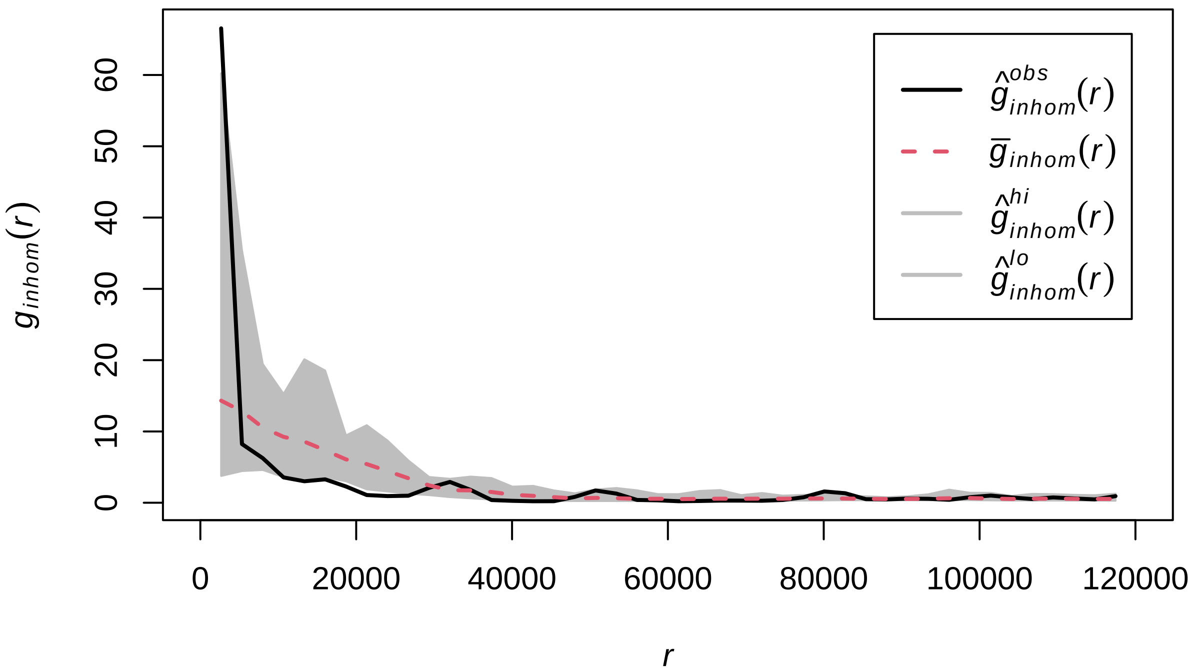


The above estimator also assumes homogeneity. Let's relax this assumption to produce a more accurate analysis.

```r
#Simulation envelope (with points drawn from the estimated intensity)
pcf_asc_inhom <- envelope(asc_data_ppp,
                          pcfinhom,
                          simulate = expression(rpoispp(lambda_asc_pos)),
                          rank = 1,
                          nsim = 19)
```

```
## Generating 19 simulations by evaluating expression  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,  19.
##
## Done.
```

```r
plot(pcf_asc_inhom,
     xlim = c(0,120000),
     main = "",
     lwd = 2)
```



Finding :

- When corrected for homogeneity, the locations of the fungi appear not to have any significant correlations.

2.3.8 Model Fitting and Selection with AIC

In this section, we will proceed with model fitting and selection, in order to achieve a reasonably effective model, while following the rule of parisomny as much as possible.

```
fit <- ppm(asc_data_ppp ~ Elevation + Dist_Water + Forest, data = BC)
fit
```

```
## Nonstationary Poisson process
## Fitted to point pattern dataset 'asc_data_ppp'
##
## Log intensity:  ~Elevation + Dist_Water + Forest
##
## Fitted trend coefficients:
##   (Intercept)     Elevation    Dist_Water        Forest
## -1.554979e+01 -8.296173e-03  7.447185e-05 -1.888729e-02
##
##                   Estimate         S.E.       CI95.lo       CI95.hi Ztest
## (Intercept) -1.554979e+01 4.578366e-02 -1.563952e+01 -1.546006e+01   ***
## Elevation   -8.296173e-03 1.475849e-04 -8.585434e-03 -8.006912e-03   ***
```

```
## Dist_Water   7.447185e-05 1.070499e-05  5.349046e-05  9.545325e-05   ***
## Forest      -1.888729e-02 8.248117e-04 -2.050390e-02 -1.727069e-02   ***
##                   Zval
## (Intercept) -339.636224
## Elevation    -56.212900
## Dist_Water     6.956742
## Forest       -22.898918
## Problem:
##  Values of the covariate 'Elevation' were NA or undefined at 0.02% (1 out of
## 5445) of the quadrature points
```

Elevation and Forest are significant but Distance to Water isn't. OK, then try one higher order.

```
fit <- ppm(asc_data_ppp ~ Elevation + I(Elevation^2) + Dist_Water + I(Dist_Water^2), data = BC)
fit
```

```
## Nonstationary Poisson process
## Fitted to point pattern dataset 'asc_data_ppp'
##
## Log intensity:  ~Elevation + I(Elevation^2) + Dist_Water + I(Dist_Water^2)
##
## Fitted trend coefficients:
##     (Intercept)       Elevation  I(Elevation^2)      Dist_Water I(Dist_Water^2)
##   -1.631404e+01   -1.100693e-02    3.133749e-06    2.916318e-04   -2.970682e-08
##
##                      Estimate         S.E.       CI95.lo       CI95.hi Ztest
## (Intercept)     -1.631404e+01 3.857234e-02 -1.638964e+01 -1.623844e+01   ***
## Elevation       -1.100693e-02 1.903877e-04 -1.138008e-02 -1.063377e-02   ***
## I(Elevation^2)   3.133749e-06 8.136546e-08  2.974275e-06  3.293222e-06   ***
## Dist_Water       2.916318e-04 3.401260e-05  2.249683e-04  3.582952e-04   ***
## I(Dist_Water^2) -2.970682e-08 4.970199e-09 -3.944823e-08 -1.996541e-08   ***
##                       Zval
## (Intercept)     -422.946643
## Elevation        -57.813223
## I(Elevation^2)    38.514484
## Dist_Water         8.574228
## I(Dist_Water^2)   -5.976989
## Problem:
##  Values of the covariate 'Elevation' were NA or undefined at 0.02% (1 out of
## 5445) of the quadrature points
##
## *** Fitting algorithm for 'glm' did not converge ***
```

Similarly, after several rounds of tuning (the details are not shown here due to repetitive and routine nature), the following seems to be a improved fit :

```
fit_simple <- ppm(asc_data_ppp ~ Elevation + Forest, data = BC)
fit_simple
```

```
## Nonstationary Poisson process
## Fitted to point pattern dataset 'asc_data_ppp'
##
## Log intensity:  ~Elevation + Forest
```

```
##
## Fitted trend coefficients:
##   (Intercept)     Elevation        Forest
## -15.476799199  -0.008114684  -0.019052930
##
##                   Estimate         S.E.       CI95.lo       CI95.hi Ztest
## (Intercept) -15.476799199 0.0440871713 -15.563208467 -15.390389931   ***
## Elevation    -0.008114684 0.0001437842  -0.008396495  -0.007832872   ***
## Forest       -0.019052930 0.0008343450  -0.020688217  -0.017417644   ***
##                   Zval
## (Intercept) -351.04995
## Elevation    -56.43657
## Forest       -22.83579
## Problem:
##  Values of the covariate 'Elevation' were NA or undefined at 0.02% (1 out of
## 5445) of the quadrature points
```

```
fit <- ppm(asc_data_ppp ~ Elevation + I(Elevation^2) + Forest + I(Forest^2), data = BC)
fit
```

```
## Nonstationary Poisson process
## Fitted to point pattern dataset 'asc_data_ppp'
##
## Log intensity:  ~Elevation + I(Elevation^2) + Forest + I(Forest^2)
##
## Fitted trend coefficients:
##    (Intercept)      Elevation I(Elevation^2)         Forest    I(Forest^2)
##  -1.510693e+01  -1.021277e-02   2.745175e-06  -3.940557e-02   2.391598e-04
##
##                     Estimate         S.E.       CI95.lo       CI95.hi Ztest
## (Intercept)    -1.510693e+01 5.147198e-02 -1.520781e+01 -1.500605e+01   ***
## Elevation      -1.021277e-02 1.762737e-04 -1.055826e-02 -9.867281e-03   ***
## I(Elevation^2)  2.745175e-06 9.595922e-08  2.557098e-06  2.933252e-06   ***
## Forest         -3.940557e-02 2.702553e-03 -4.470248e-02 -3.410867e-02   ***
## I(Forest^2)     2.391598e-04 2.872955e-05  1.828510e-04  2.954687e-04   ***
##                      Zval
## (Intercept)    -293.498126
## Elevation       -57.936993
## I(Elevation^2)   28.607725
## Forest          -14.580871
## I(Forest^2)       8.324525
## Problem:
##  Values of the covariate 'Elevation' were NA or undefined at 0.02% (1 out of
## 5445) of the quadrature points
##
## *** Fitting algorithm for 'glm' did not converge ***
```

Now, the model gets more complicated. We will evaluate if the additional cost overhead of adopting such a more complicated model is well justified by benchmarking with the AIC values.

```
#AIC values
AIC(fit); AIC(fit_simple)
```

```
## [1] 64029.39
```

```
## [1] 64238.64
```

```
#Delta AIC
AIC(fit_simple) - AIC(fit)
```

```
## [1] 209.2528
```

We will also like to conduct a anova LRT test as an additional objective measurement to compare the two models :

```
anova(fit_simple, fit, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: ~Elevation + Forest      Poisson
## Model 2: ~Elevation + I(Elevation^2) + Forest + I(Forest^2)   Poisson
##   Npar Df Deviance  Pr(>Chi)
## 1    3
## 2    5  2   213.25 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The following conclusion is drawn :

- The model with quadratic terms provides a better fit to the data
- With the delta AIC of 44, the extra complexity is well supported.
- So, a possible model should be

$$\lambda_{ASC}(u) = e^{-16.2\ -\ 0.017\ \text{elevation}(u)\ -\ 0.000003\ \text{elevation}(u)^2\ -\ 0.042\ \text{forest}(u)\ +\ 0.00018\ \text{forest}(u)^2}$$

2.3.9 Model Validation

Now, given the following fitted model, we will finally evaluate the over model performance again with Quadrat Test and PPP Residuals.

$$\lambda_{ASC}(u) = e^{-16.2\ -\ 0.017\ \text{elevation}(u)\ -\ 0.000003\ \text{elevation}(u)^2\ -\ 0.042\ \text{forest}(u)\ +\ 0.00018\ \text{forest}(u)^2}$$

2.3.9.a Quadrat Test Let's see the quadrat test result of the higher degree polynomial model.

```
#Run the quadrat test
quadrat.test(fit, nx = 4, ny = 4)
```

```
##
##  Chi-squared test of fitted Poisson model 'fit' using quadrat counts
##
## data:  data from fit
## X2 = 2534.5, df = 8, p-value < 2.2e-16
## alternative hypothesis: two.sided
##
## Quadrats: 13 tiles (irregular windows)
```

This has small p value, suggesting significant deviation from our model's prediction. Room for further improvement is therefore expected, but it does not provide hint for how to achieve any improvement.

2.3.9.b PPP Residuals (SKIPPED AS RETURN ERROR)

We will also check the distribution of residuals value of the model.

From the plot, it's observed that :

- The residuals are small in magnitude (e^-8).

- The negative residual values suggest over-prediction of the model.
- There is room for improvement for the model

2.3.10 Higher Order Polynomial Fitting with Spline and Validation

As the above analysis indicated a room for improvement, we will finally try to add higher-order polynomials with the spline packages.

```
library(splines)

#Fit the PPP model
fit_smooth <- ppm(asc_data_ppp ~ bs(Elevation,7) + bs(Forest, 3), data = BC, use.gam = TRUE)

fit_smooth
```

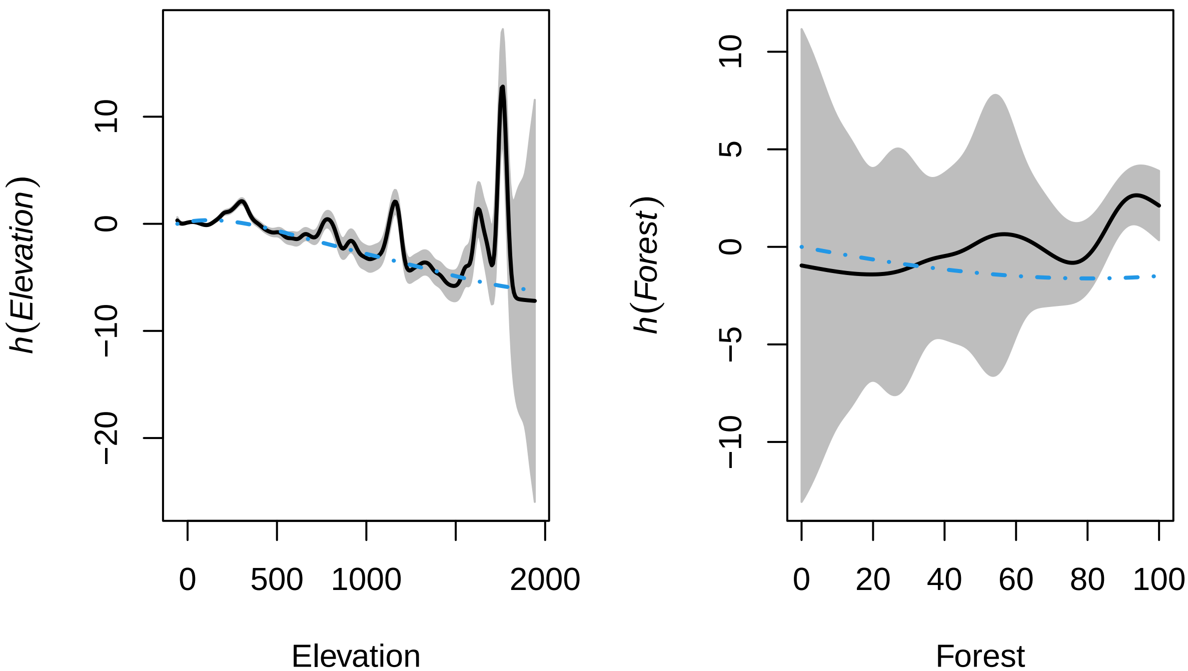
```
## Nonstationary Poisson process
## Fitted to point pattern dataset 'asc_data_ppp'
##
## Log intensity:  ~bs(Elevation, 7) + bs(Forest, 3)
##
## Fitted trend coefficients:
##       (Intercept) bs(Elevation, 7)1 bs(Elevation, 7)2 bs(Elevation, 7)3
##        -15.704714          0.933708         -1.403418         -4.191303
## bs(Elevation, 7)4 bs(Elevation, 7)5 bs(Elevation, 7)6 bs(Elevation, 7)7
##         -7.303928         -4.163338        -31.646034         -7.398845
##    bs(Forest, 3)1    bs(Forest, 3)2    bs(Forest, 3)3
##         -1.167421         -1.928981         -1.492493
##
## For standard errors, type coef(summary(x))
## Problem:
##  Values of the covariate 'Elevation' were NA or undefined at 0.02% (1 out of
## 5445) of the quadrature points
```

```
#Calculate the partial residuals as a function of elevation
par_res_elev <- parres(fit_smooth, "Elevation")

#Calculate the relative intensity as a function of gradient
par_res_forest <- parres(fit_smooth, "Forest")

#Side by side plotting
par(mfrow = c(1,2))
plot(par_res_elev,
     legend = FALSE,
```

```
     lwd = 2,
     main = "",
     xlab = "Elevation")
plot(par_res_forest,
     legend = FALSE,
     lwd = 2,
     main = "",
     xlab = "Forest")
```



Now, compare the AIC values for both the simpler and higher polynomial degree models.

```
#AIC values
AIC(fit); AIC(fit_smooth)
```

```
## [1] 64029.39
```

```
## [1] 63993.06
```

```
#Delta AIC
AIC(fit) - AIC(fit_smooth)
```

```
## [1] 36.32074
```

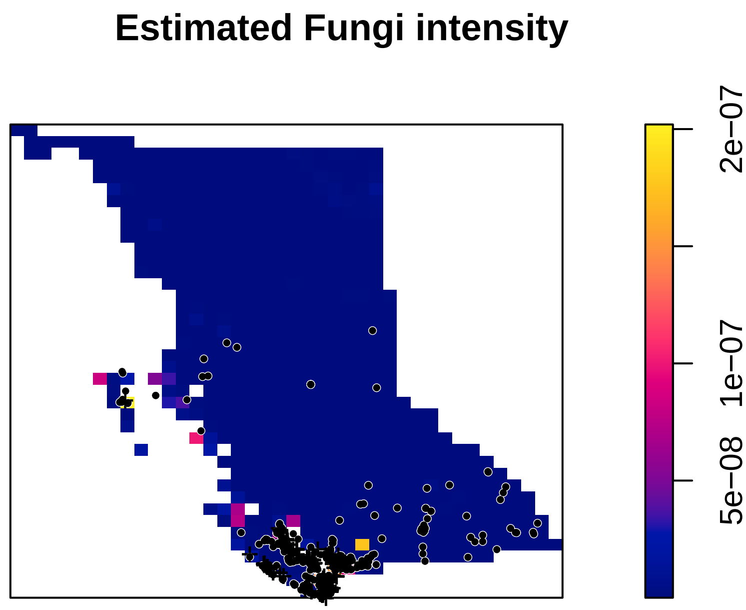
```
#Likelihood ratio test
anova(fit, fit_smooth, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: ~Elevation + I(Elevation^2) + Forest + I(Forest^2)   Poisson
## Model 2: ~bs(Elevation, 7) + bs(Forest, 3)    Poisson
##   Npar Df Deviance  Pr(>Chi)
## 1    5
## 2   11  6   48.321 1.019e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All these suggest that this complex models provides a better fit to the data. Let's finally visualize the predictions as before.

```
#Plot the model predictions
plot(fit_smooth,
     se = FALSE,
     superimpose = FALSE,
     main = "Estimated Fungi intensity")

#Overlay the ocations
plot(asc_data_ppp,
     pch = 16,
     cex = 0.6,
     cols = "white",
     add = TRUE)
plot(asc_data_ppp,
     pch = 16,
     cex = 0.5,
     cols = "black",
     add = TRUE)
```



From this visualisation, the following is observed :

- Although the model is not yet perfect, it is progressively having improvement after rounds of variables selection process.

- Considering the fact that we are predicting the locations of one species of fungi in a biodiverse continent based only on Elevation and Forest, and have no information on all of the many other factors that would significantly influence fungi growth (e.g. humidity, moisture level, temperature, pH value, oxygen content, etc.)

3. Discussion:

Provide a brief summary of your findings. Length: ca. 1 page.

4. References:

Include references to all necessary literature.