

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 - Lichen 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") ## 102 illegal points stored in attr(",r

# 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 columns 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 is shown below.

```

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

```

```

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.

```
plot(asc_data_ppp)
```

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 occurrences 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.

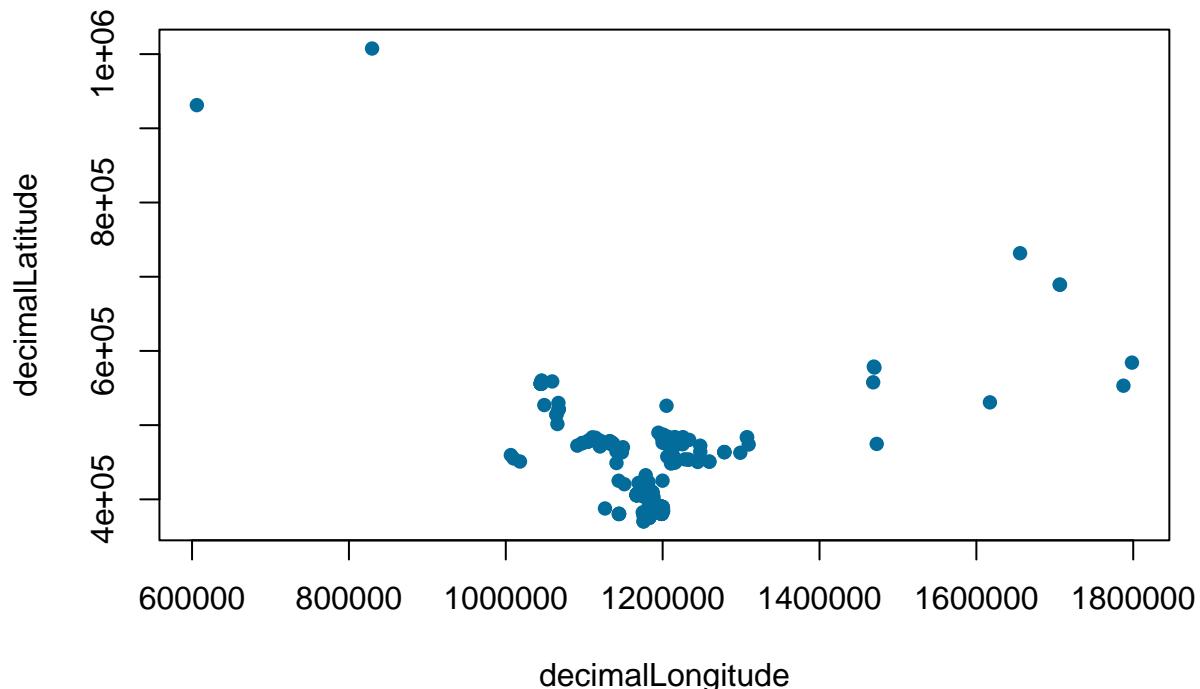
We will then conduct First Moment Analysis which will include homogeneity analysis and covariates study on some variables. Afterwards, we will perform Second Moment analysis where popular tools such as K-function and Pair Correlation Function will be deployed to investigate any clustering.

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 Walkthrough

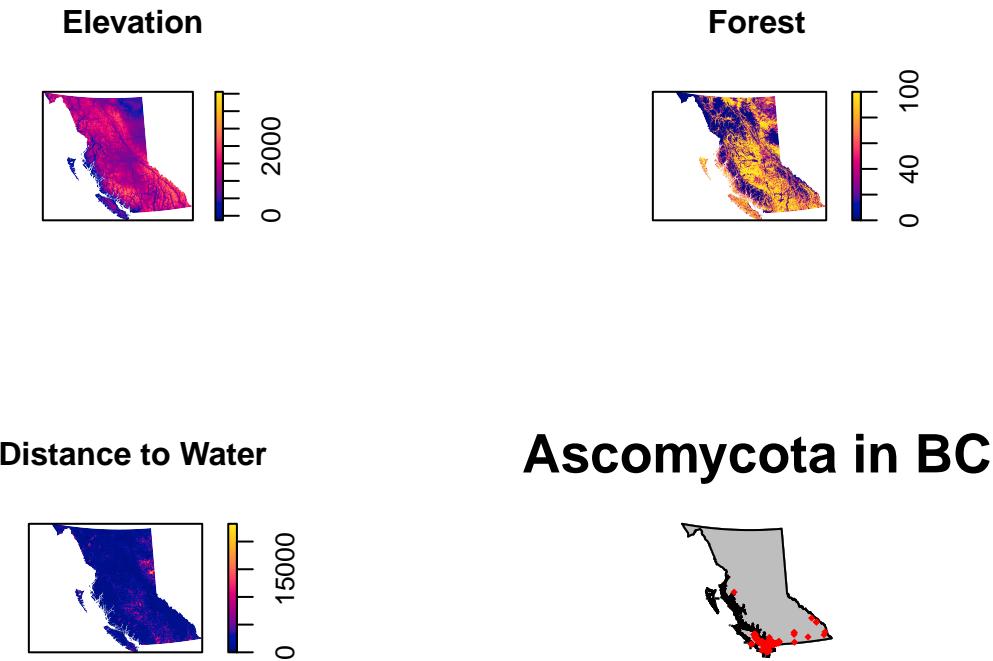
Let's see what the BC_Covariates.Rda file provide us which may help us to identify and choose appropriate fields to be used in the covariates analysis in the later part of this analysis. It should contain the information on the BC province :

- 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
```



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
##      315          8            1            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] 91.271680 91.271680 23.397900 62.692406 51.428791 261.980282
## [7] 261.980282 261.980282 74.498003 74.498003 510.926444 32.118248
## [13] 62.814557 81.548294 81.548294 76.084493 706.409676 65.194619
```

```

## [19] 8.000000 -35.179669 -35.179669 -35.179669 -35.179669 76.084493
## [25] 76.084493 76.084493 32.118248 32.118248 195.284587 61.665185
## [31] 206.588250 66.000000 65.619479 288.070603 288.070603 288.070603
## [37] 288.070603 288.070603 355.967892 355.967892 355.967892 355.967892
## [43] 355.967892 355.967892 355.967892 288.070603 288.070603 288.070603
## [49] 288.070603 288.070603 288.070603 288.070603 288.070603 288.070603
## [55] 288.070603 288.070603 430.115273 430.115273 288.070603 288.070603
## [61] 355.967892 430.115273 430.115273 288.070603 288.070603 288.070603
## [67] 288.070603 288.070603 288.070603 225.481611 225.481611 198.143926
## [73] 81.126810 -35.053897 -35.053897 39.369451 39.369451 294.384024
## [79] -35.053897 -35.053897 -35.053897 -35.053897 -35.053897 -35.053897
## [85] 71.801786 -35.053897 187.957696 187.957696 171.936613 167.107333
## [91] 40.396331 40.396331 40.396331 40.396331 40.396331 40.396331
## [97] 1038.434881 777.098704 1670.964740 69.289789 94.422608 69.289789
## [103] 39.092076 82.583477 14.698512 55.883115 36.396924 49.126893
## [109] 45.652949 50.701483 50.701483 50.784868 49.126893 44.246494
## [115] 25.428184 25.428184 25.428184 339.591607 21.000000 21.000000
## [121] 21.000000 25.141153 1119.185852 108.960033 45.652949 32.118248
## [127] 23.899560 92.143706 42.880855 54.069165 50.784868 130.735146
## [133] 147.005958 50.784868 1318.635190 131.236704 114.939521 131.236704
## [139] 160.090881 44.246494 1130.775108 46.432839 -3.734969 20.714393
## [145] 44.000000 122.489656 53.985568 177.952547 46.432839 117.749693
## [151] 50.784868 46.432839 72.717695 433.194915 33.995941 72.717695
## [157] 72.717695 48.896700 25.547370 55.243721 55.243721 2.895689
## [163] 32.118248 197.677291 46.440137 46.440137 71.801786 198.143926
## [169] 23.397900 32.118248 80.825577 102.071038 433.786763 224.330101
## [175] 224.330101 189.297238 214.439717 189.990055 203.326184 203.326184
## [181] 203.326184 203.326184 203.326184 206.588250 22.034083 18.242945
## [187] -35.053897 45.657363 510.926444 510.926444 197.677291 197.677291
## [193] -57.458214 60.833698 50.784868 206.588250 -35.053897 -35.053897
## [199] 32.118248 181.306638 27.681348 27.681348 176.437454 176.437454
## [205] 37.000000 32.118248 46.432839 26.520313 62.814557 4.412054
## [211] 315.807853 315.807853 40.609456 135.806165 50.784868 66.741829
## [217] 31.212766 53.507950 267.141364 233.374089 129.407323 197.677291
## [223] 1119.185852 67.901429 145.876827 235.703330 334.238754 1108.387686
## [229] 206.488760 206.488760 206.488760 193.400186 115.019727 32.118248
## [235] 27.407411 32.118248 32.118248 32.118248 32.118248 44.246494
## [241] 44.246494 26.520313 26.520313 26.520313 26.520313 26.520313
## [247] 26.520313 26.520313 26.520313 26.520313 26.520313 26.520313
## [253] 26.520313 26.520313 26.520313 26.520313 26.520313 404.858785
## [259] 26.520313 26.520313 26.520313 26.520313 26.520313 26.520313
## [265] 26.520313 26.520313 26.520313 26.520313 26.520313 26.520313
## [271] 26.520313 26.520313 32.118248 32.118248 43.852612 43.852612
## [277] 43.852612 26.520313 26.520313 26.520313 26.520313 26.520313
## [283] 26.520313 26.520313 26.520313 26.520313 26.520313 26.520313
## [289] 26.520313 26.520313 26.520313 26.520313 26.520313 26.520313
## [295] 26.520313 26.520313 26.520313 26.520313 26.520313 26.520313
## [301] 26.520313 26.520313 26.520313 26.520313 26.520313 26.520313
## [307] 26.520313 26.520313 26.520313 26.520313 62.814557 62.814557
## [313] 62.814557 62.814557 62.814557 64.156157 23.804111 45.652949
## [319] 26.520313 36.396924 -6.244630 123.164317 205.363512 205.363512

```

```

library("kdensity")
asc_density <- density(BC$Elevation[asc_data_ppp])

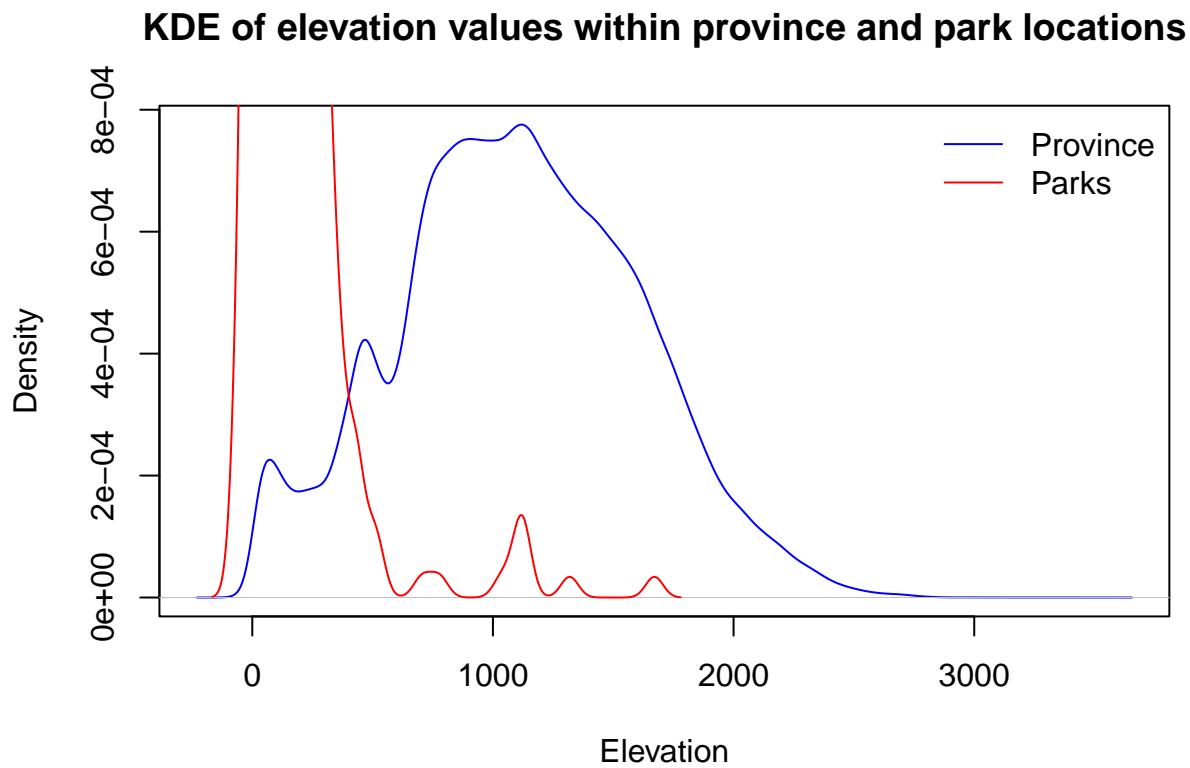
```

```

province_density <- density(BC$Elevation$v, na.rm=T)

plot(province_density, main = "KDE of elevation values within province and park locations", xlab = "Elevation"
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 = "Elevation"
# 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] 4.218242e-10
```

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 sparsely located in the BC. Let's verify this with the Quadrat Count Plot.

```

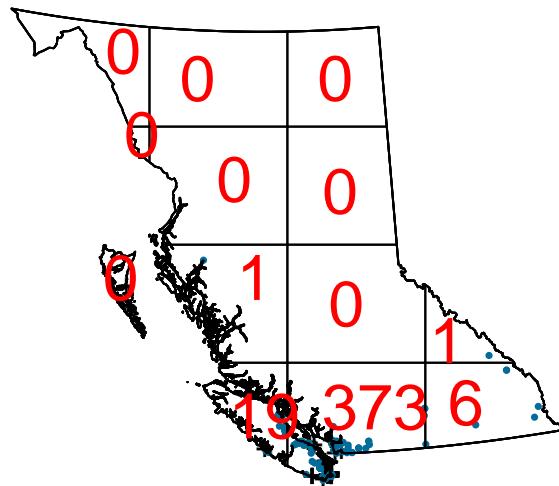
Q <- quadratcount(asc_data_ppp,
                     nx = 4,
                     ny = 4)

#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)

```

Ascomycota Locations – Quadrat Count



```

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

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

```

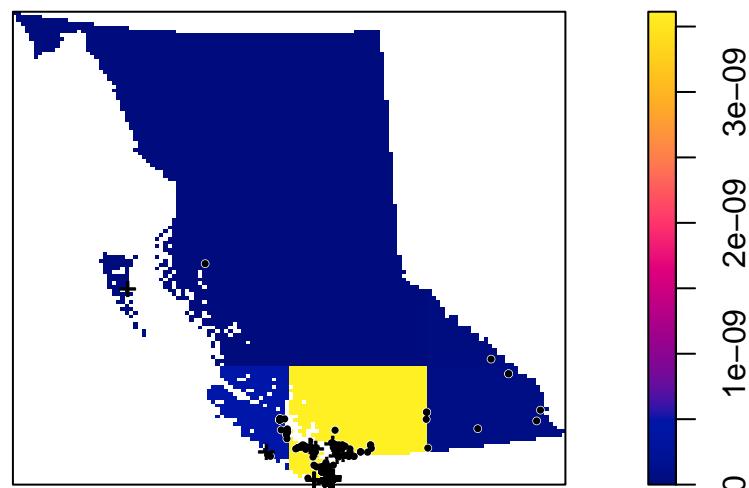
```

add = T)

plot(asc_data_ppp,
      pch = 16,
      cex = 0.5,
      cols = "black",
      add = T)

```

Ascomycota intensity



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 = 2818.1, 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)

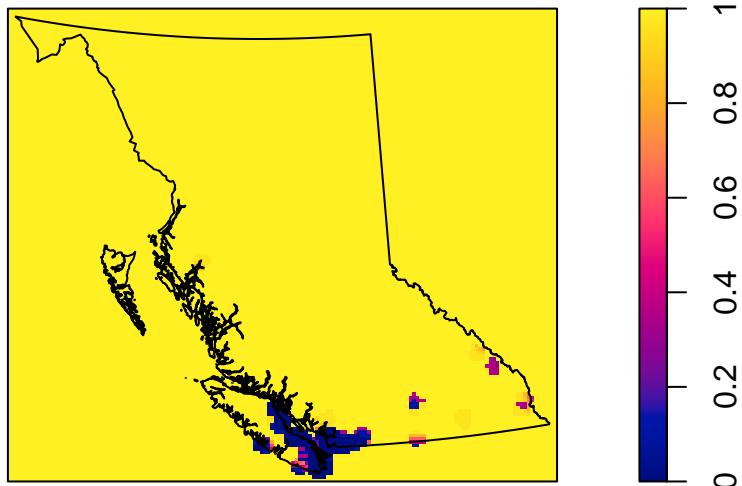
#plot(LR, "Likelihood Ratio Test")
#plot(asc_data_ppp$window, "Likelihood Ratio Test")

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)

```

Local p-values



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 yellow (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

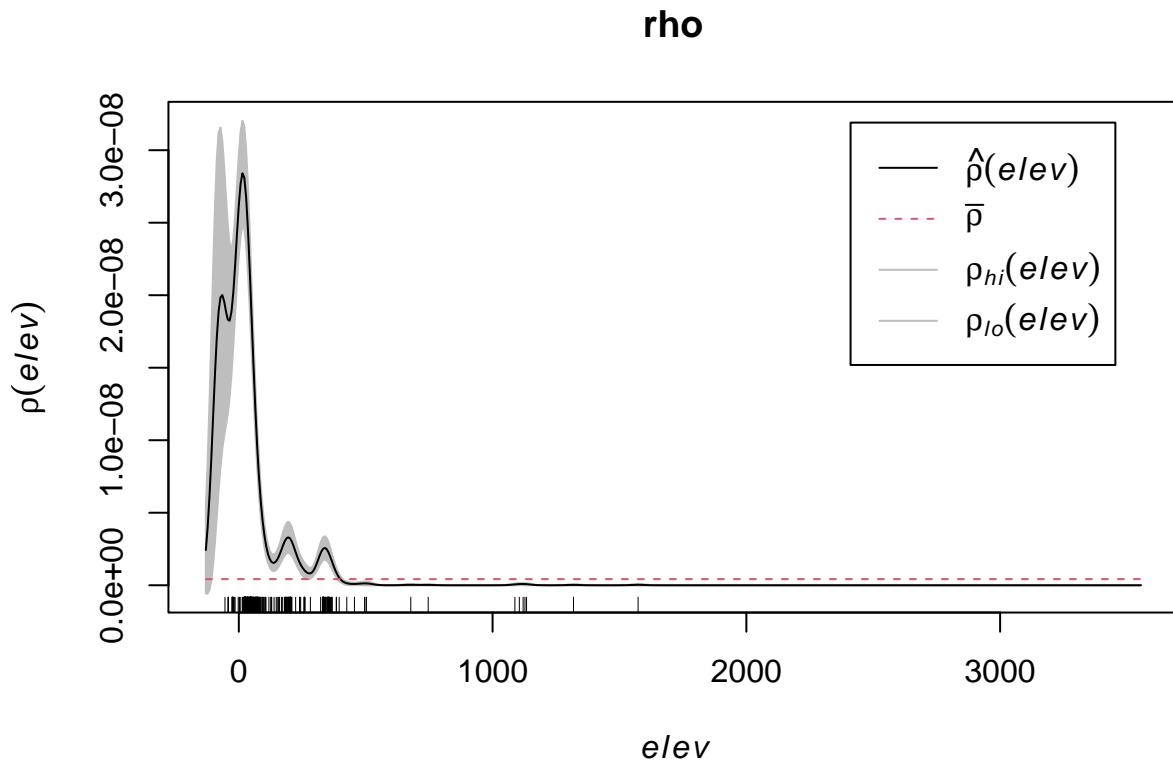
Let's start with Elevation.

```

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]
##          392                  2                 5                  1

```



2.3.6.b Covariate Variable : Forest

Then, let's see Forest.

```

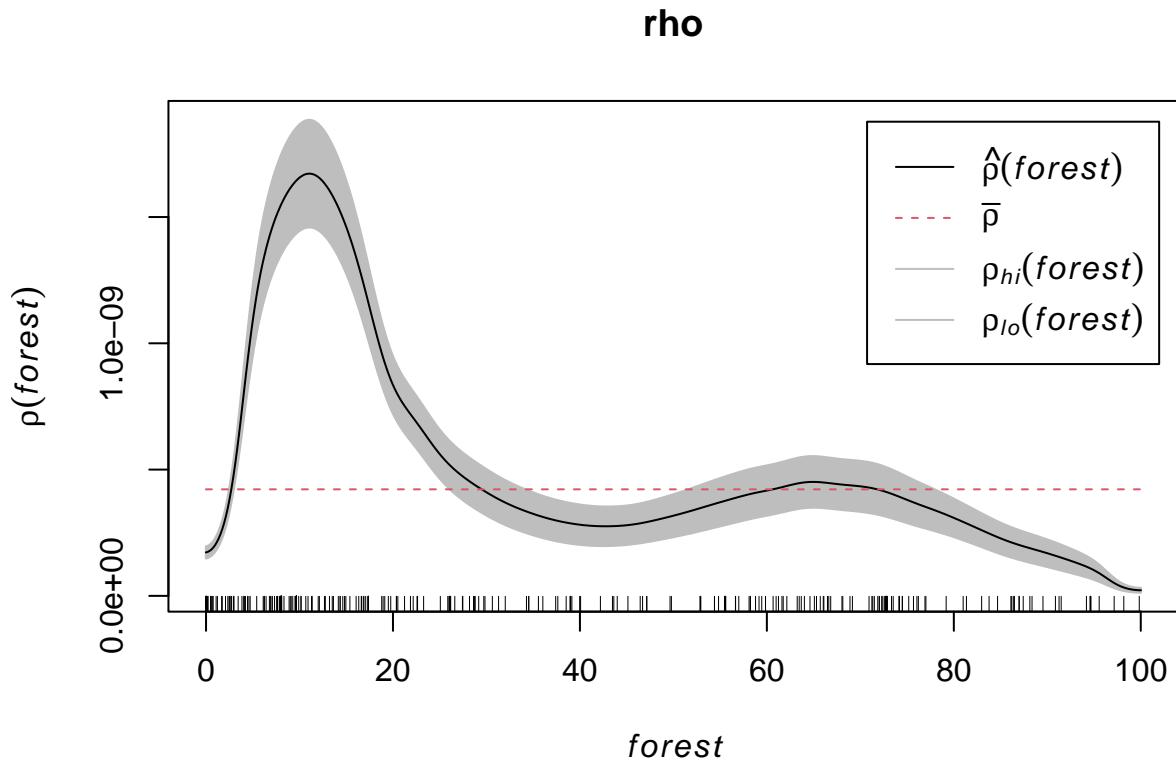
forest <- BC$Forest
b <- quantile(forest,probs=(0:4)/4,type=2)
Zcut <- cut(forest,breaks=b)
V <- tess(image=Zcut)
quadratcount(asc_data_ppp,tess=V)

```

```

## tile
##      (0,11.6] (11.6,50.2] (50.2,86.9] (86.9,100]
##      197         93        81        29

```



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

```

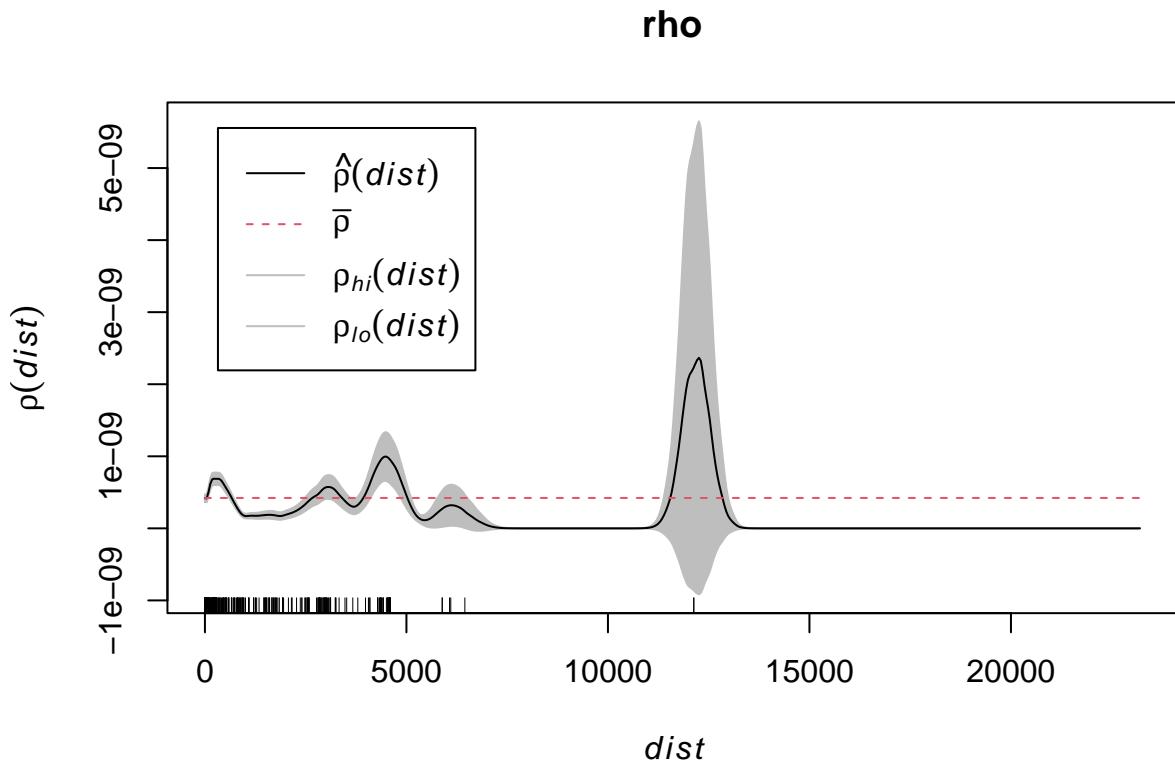
dist <- BC$Dist_Water
b <- quantile(dist,probs=(0:4)/4,type=2)
Zcut <- cut(dist,breaks=b)
V <- tess(image=Zcut)
quadratcount(asc_data_ppp,tess=V)

```

```

## tile
##      (0,483]      (483,1.1e+03]   (1.1e+03,2.18e+03] (2.18e+03,2.32e+04]
##      213           47            38            96

```



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.

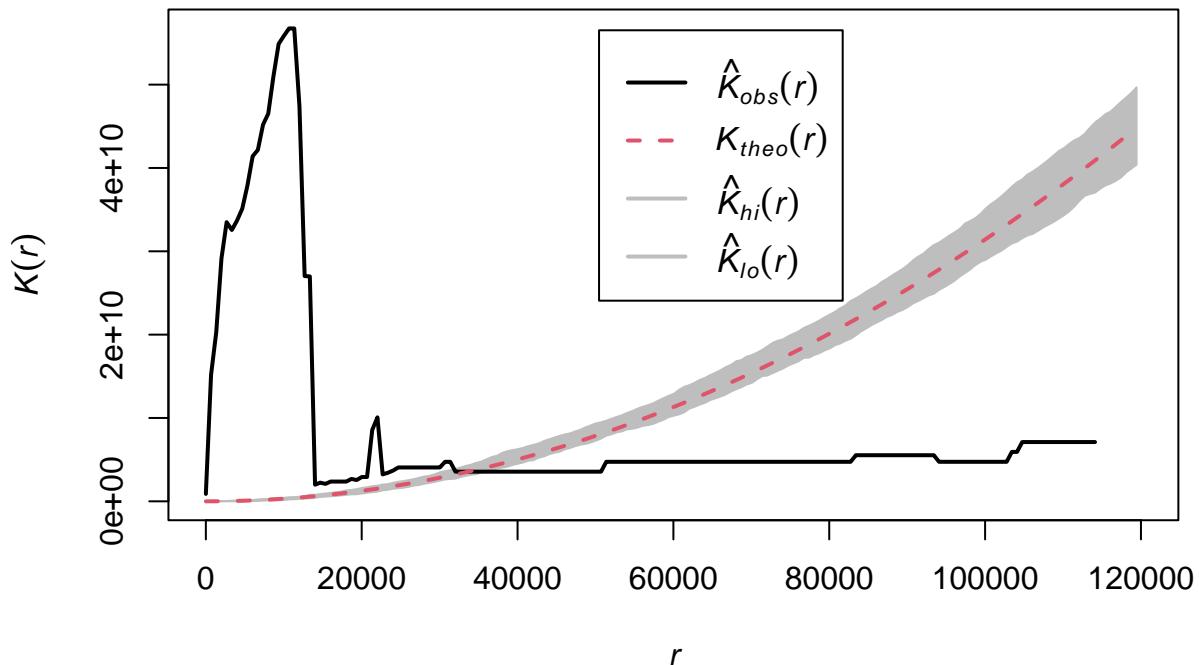
```

# 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.

# 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.

```

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)

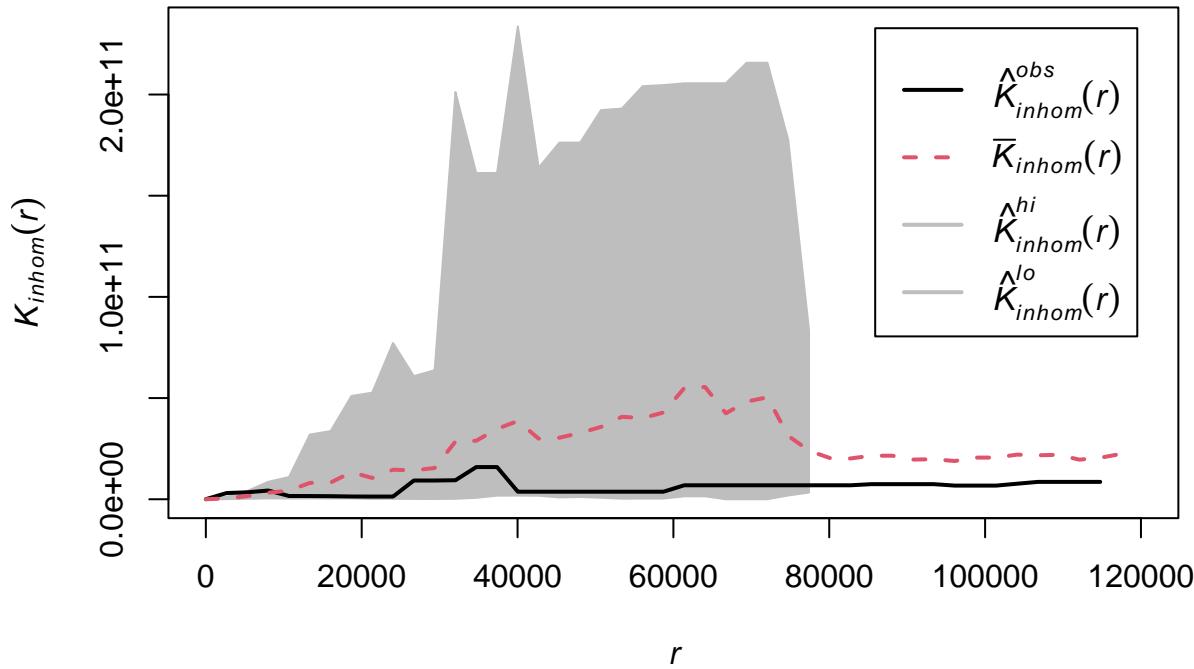
#Simulation envelope (with points drawn from the estimated intensity)
E_asc_inhom <- envelope(asc_data_ppp,
                         Kinhom,
                         simulate = expression(rpoispp(lambda_asc_pos)),
                         correction="border",
                         rank = 1,
     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

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

```

Ripley's K-function (Inhomogeneity)



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.

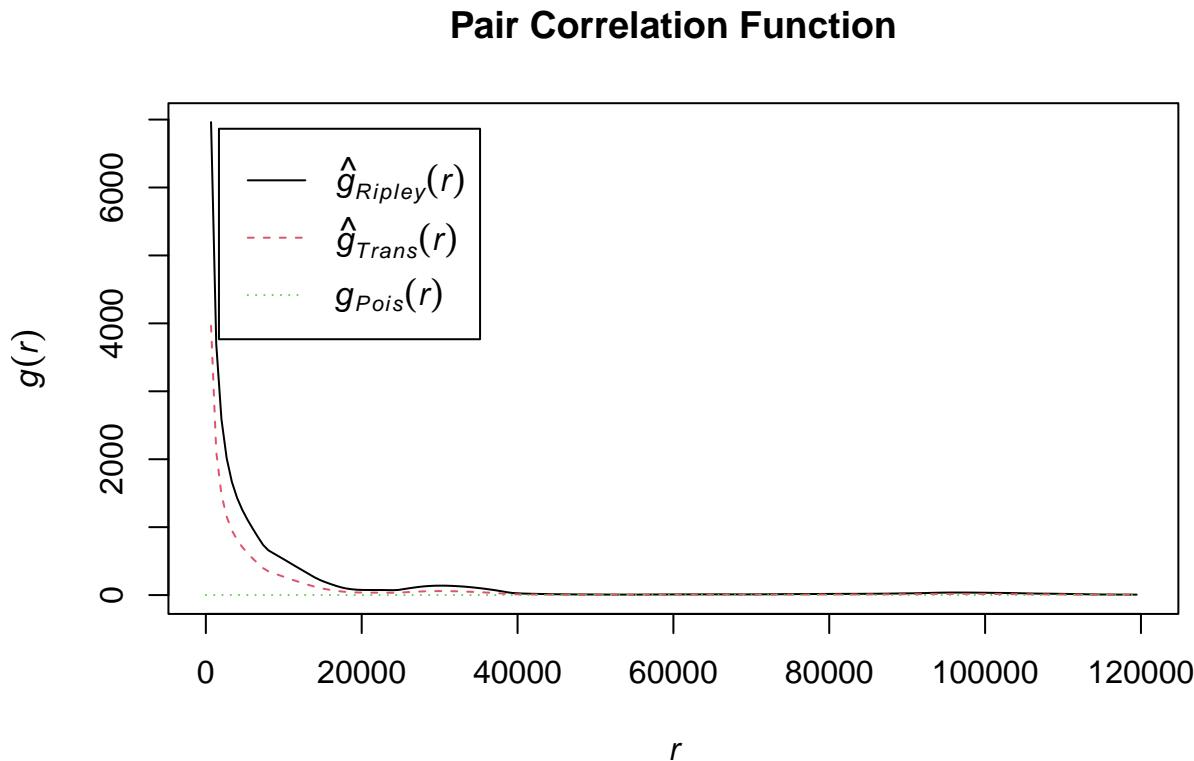
2.3.7.b Pair Correlation Function We are also interested in checking the points relation using pair correlation function.

```
# 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              distance argument r
##      theo            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]
```

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

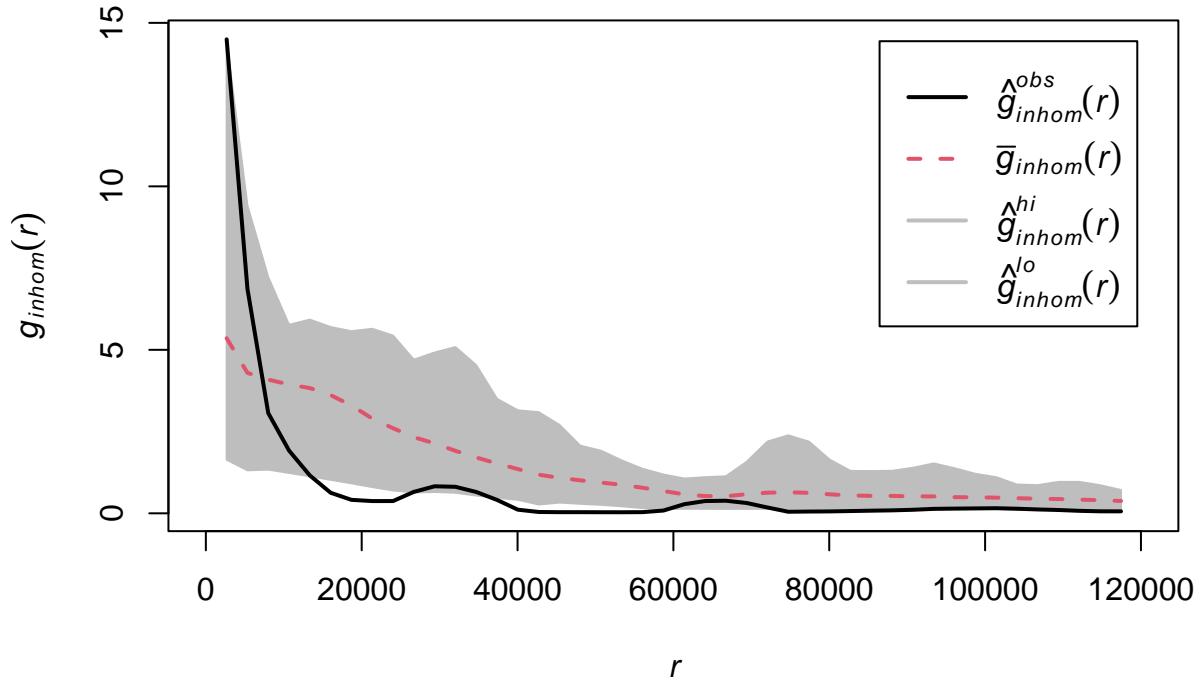


The above estimator also assumes homogeneity. Let's relax this assumption to produce a more accurate analysis.

```
#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.
```

```
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 parsimony 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
## -1.649783e+01 -9.746289e-03  5.074702e-05 -2.772079e-02
##
##             Estimate       S.E.     CI95.lo     CI95.hi Ztest
## (Intercept) -1.649783e+01 8.719049e-02 -1.666872e+01 -1.632694e+01 *** 
## Elevation   -9.746289e-03 3.872771e-04 -1.050534e-02 -8.987240e-03 ***
```

```

## Dist_Water  5.074702e-05 2.503222e-05  1.684771e-06  9.980928e-05      *
## Forest      -2.772079e-02 1.803605e-03 -3.125579e-02 -2.418579e-02    ***
##                         Zval
## (Intercept) -189.215943
## Elevation   -25.166190
## Dist_Water   2.027268
## Forest      -15.369657
## *** Fitting algorithm for 'glm' did not converge ***

```

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)
## -1.751444e+01 -1.298691e-02  3.757783e-06  2.313973e-04 -2.416955e-08
##
##                         Estimate       S.E.      CI95.lo      CI95.hi Ztest
## (Intercept) -1.751444e+01 8.075694e-02 -1.767272e+01 -1.735616e+01 *** 
## Elevation   -1.298691e-02 5.148358e-04 -1.399597e-02 -1.197785e-02 *** 
## I(Elevation^2) 3.757783e-06 1.849431e-07  3.395301e-06  4.120264e-06 *** 
## Dist_Water   2.313973e-04 7.552735e-05  8.336647e-05  3.794282e-04 ** 
## I(Dist_Water^2) -2.416955e-08 1.138818e-08 -4.648997e-08 -1.849130e-09 * 
##
##                         Zval
## (Intercept) -216.878412
## Elevation   -25.225346
## I(Elevation^2) 20.318594
## Dist_Water   3.063756
## I(Dist_Water^2) -2.122337
## *** 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
## -16.449678447 -0.009601581 -0.027922868
##
##                         Estimate       S.E.      CI95.lo      CI95.hi Ztest

```

```

## (Intercept) -16.449678447 0.0831309100 -16.61261204 -16.286744858 ***  

## Elevation -0.009601581 0.0003758118 -0.01033816 -0.008865003 ***  

## Forest -0.027922868 0.0018099253 -0.03147026 -0.024375480 ***  

## Zval  

## (Intercept) -197.87680  

## Elevation -25.54891  

## Forest -15.42764  

## *** Fitting algorithm for 'glm' did not converge ***  

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)  

## -1.621033e+01 -1.173001e-02 3.259440e-06 -4.196879e-02 1.826759e-04  

##  

## Estimate S.E. CI95.lo CI95.hi Ztest  

## (Intercept) -1.621033e+01 9.367906e-02 -1.639394e+01 -1.602673e+01 ***  

## Elevation -1.173001e-02 4.272309e-04 -1.256736e-02 -1.089265e-02 ***  

## I(Elevation^2) 3.259440e-06 1.877594e-07 2.891438e-06 3.627441e-06 ***  

## Forest -4.196879e-02 5.890907e-03 -5.351476e-02 -3.042283e-02 ***  

## I(Forest^2) 1.826759e-04 6.812603e-05 4.915130e-05 3.162004e-04 **  

## Zval  

## (Intercept) -173.041168  

## Elevation -27.455891  

## I(Elevation^2) 17.359658  

## Forest -7.124334  

## I(Forest^2) 2.681440  

## *** 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] 15464.31
```

```
## [1] 15508.45
```

```
#Delta AIC  

AIC(fit_simple) - AIC(fit)
```

```
## [1] 44.14666
```

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  48.147 3.508e-11 ***
## ---
## 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 = 585.05, 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 We will also check the distribution of residuals value of the model.

```
#Calculate the residuals
res <- residuals(fit)

#Visualise
plot(res,
#      cols = "transparent",
main="Residuals Plots")
```

Residuals Plots



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
```

```

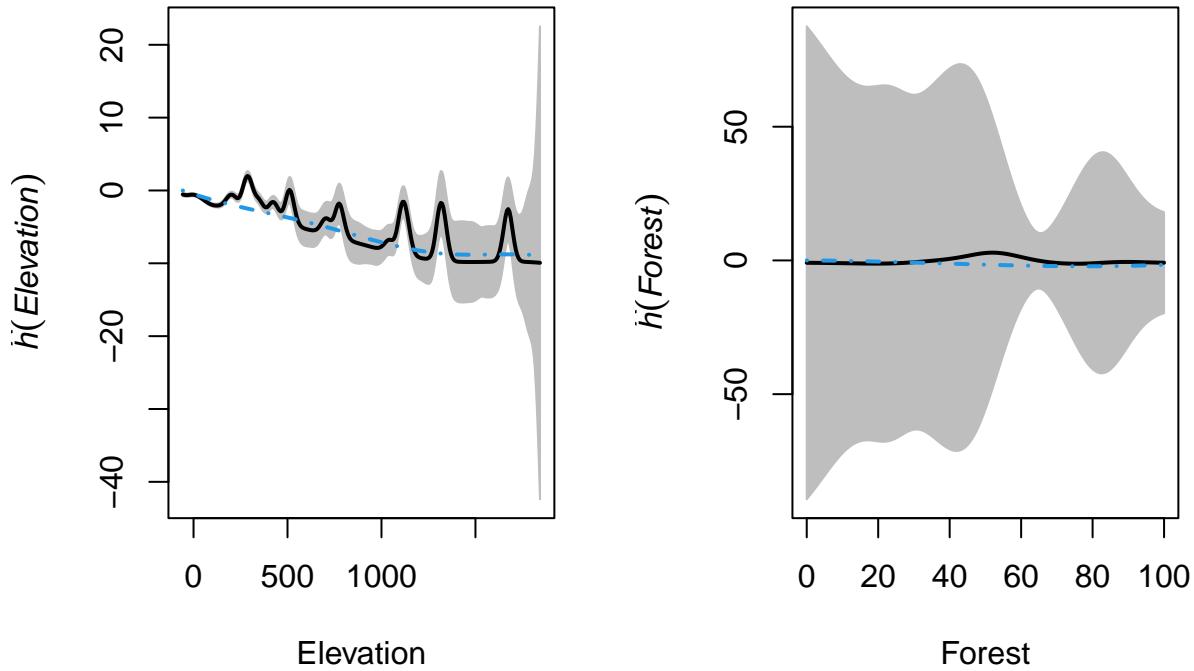
## Nonstationary Poisson process
## Fitted to point pattern dataset 'asc_data_ppp'
##
## Log intensity: ~bs(Elevation, 7) + bs(Forest, 3)
##
## Fitted trend coefficients:
##      (Intercept) bs(Elevation, 7)1 bs(Elevation, 7)2 bs(Elevation, 7)3
## -15.1837101     -2.1215999     -3.9107238     -10.2916364
## bs(Elevation, 7)4 bs(Elevation, 7)5 bs(Elevation, 7)6 bs(Elevation, 7)7
## -6.9600818     -20.1516125     49.1493840    -1129.0583122
##   bs(Forest, 3)1   bs(Forest, 3)2   bs(Forest, 3)3
## -0.2110034     -3.4557922     -1.6215046
##
## For standard errors, type coef(summary(x))

#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] 15464.31

## [1] 15440.68

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

## [1] 23.62747

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

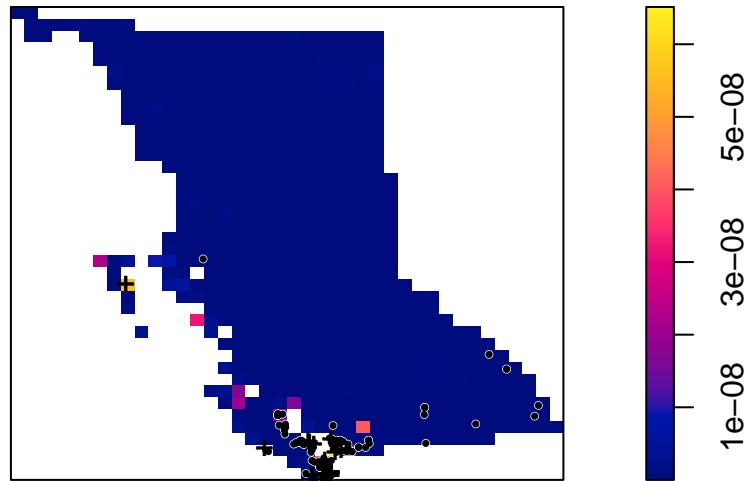
## Analysis of Deviance Table
##
## Model 1: ~Elevation + I(Elevation^2) + Forest + I(Forest^2)    Poisson
## Model 2: ~bs(Elevation, 7) + bs(Forest, 3)    Poisson
##   Npar Df Deviance  Pr(>Chi)
## 1     5
## 2    11  6  35.622 3.264e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 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 locations
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)
```

Estimated Fungi intensity



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.