spatial Interpolation

G. Fenton

5/20/2022

Problem 4: Sampled point data interpolated onto a 2D structured grid

Clean the data space

Load the required libraries in R for processing the data. The interpolation routine (IDW) and the point-pattern-processor routines are already pretty standard routines in the base R library space.

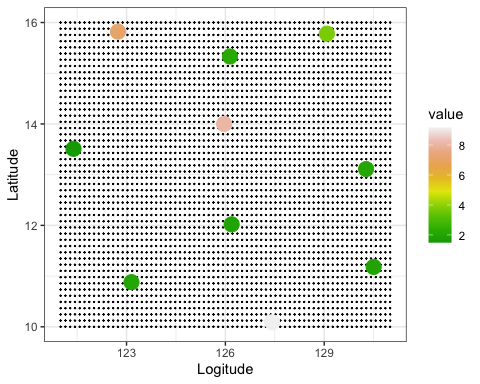
library(ggplot2)

Look at the provided data that will be interpolated onto a suggested grid prescribed below. Create the spatial.data dataframe from the provided longitude, latitude, value data.

# data points  
long = c(121.39,126.19,130.27,127.42,126.14,125.96,123.15,130.5,129.08,122.74)  
lat = c(13.51,12.02,13.11,10.09,15.33,14,10.88,11.18,15.78,15.82)  
value = c(1.494,1.934,2.148,9.155,2.221,8.1,2.039,1.916,3.729,7.137)  
  
spatial.data <- data.frame(long, lat, value)

Set up a grid template for interpolation. The suggested boundaries are a minimum of 121.0 and a maximum of 131.0 in longitude and a minimum of 10.0 and a maximum of 16.0 in latitude. Setup the bounding box based on the specifications. Plot the suggested grid with the observed data values.

bbox <- c(  
 "xmin" = 121.0,  
 "ymin" = 10.0,  
 "xmax" = 131.0,  
 "ymax" = 16.0  
)  
  
grid\_template <- expand.grid(  
 Logitude = seq(from = bbox["xmin"], to = bbox["xmax"], length.out = 70),   
 Latitude = seq(from = bbox["ymin"], to = bbox["ymax"], length.out = 50)   
)  
  
grid\_plot <- ggplot() +  
 geom\_point(data = grid\_template, aes(x = Logitude, y = Latitude), size = 0.1) +  
 geom\_point(data = spatial.data, mapping = aes(x = long, y = lat, color=value), size=5) +  
 scale\_color\_gradientn(colors = terrain.colors(10)) +  
 coord\_cartesian(xlim = c(121, 131), ylim = c(10, 16)) +  
 theme\_bw()  
  
grid\_plot

 Now we have set up our points and a grid to interpolate onto, we are ready to do an interpolation. The method I am using is the Inverse Distance Weighting (IDW). This method is reasonable and fairly simple to use as the IDW method does not require a special setup to model spatial relationships. I use this method frequently when mapping dozens of pressure gauge locations onto a topological mapping of terrain. I am routinely doing this to process dozens of time based pressure reconds consisting of hundreds of thousands of time-value data pairs. R is a great tool to have to work with a lot of large datasets.

Create the object window which is defined by the specified longitude and latitude limits. This window is the item that we will use to interpolate the spatial.data onto. Once object window is defined, the next step includes creation of point pattern object (ppp) within the specified object window. Marks are added as attribute value to each point of the spatial.data$value.

# IDW Interpolation  
library(gstat)  
library(spatstat)

## Loading required package: spatstat.data

## Loading required package: spatstat.geom

## spatstat.geom 2.4-0

## Loading required package: spatstat.random

## spatstat.random 2.2-0

## Loading required package: spatstat.core

## Loading required package: nlme

## Loading required package: rpart

## spatstat.core 2.4-4

##   
## Attaching package: 'spatstat.core'

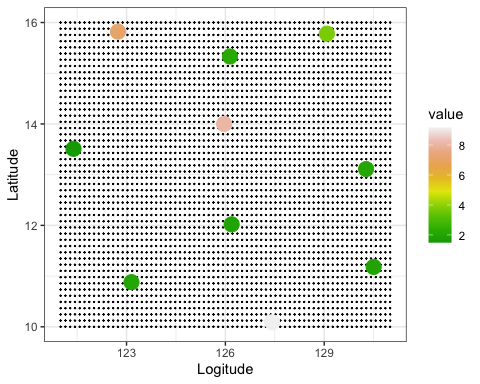
## The following object is masked from 'package:gstat':  
##   
## idw

## Loading required package: spatstat.linnet

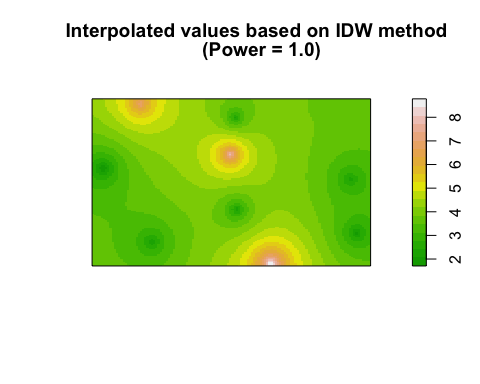
## spatstat.linnet 2.3-2

##   
## spatstat 2.3-4 (nickname: 'Watch this space')   
## For an introduction to spatstat, type 'beginner'

grid\_plot



# create the object window based on the specified grid template dimensions  
obs\_window <- owin(c(min(grid\_template$Logitude), max(grid\_template$Logitude)),  
 c(min(grid\_template$Latitude), max(grid\_template$Latitude)))  
  
# create the point-pattern-object (ppp) based on the provided data points (spatial.data)  
ppp\_z <- ppp(spatial.data$long, spatial.data$lat,   
 marks=spatial.data$value, window=obs\_window)  
  
# create the IDW data from the created point-pattern object based on the provided spatioal data  
idw\_data <- idw(ppp\_z, power=1, at="pixels", se=FALSE)  
  
# plot the IDW data using the base R plotting package. Have an issue getting the  
# axis labels to appear.  
plot(idw\_data,  
 col= terrain.colors(20),  
 xlab="Longitude",  
 ylab="Latitude",  
 main="Interpolated values based on IDW method \n (Power = 1.0)" )



Plot the same IDW data using ggplot to get a similar interpolated result with all the axis labels and plot title.

idw.output = as.data.frame(idw\_data)  
names(idw.output)[1:3]<-c("long","lat","value")  
  
plot <- ggplot(data=idw.output, aes(x=long, y=lat))  
  
layer1 <- c(geom\_tile(data=idw.output, aes(fill=value)))  
  
plot + layer1 +   
 geom\_raster(aes(fill = value)) +  
 scale\_fill\_gradientn(colors = terrain.colors(20)) +  
 #scale\_fill\_distiller(palette = "Spectral") +  
 coord\_equal() +   
 labs(title="Interpolated values based on IDW method \n (Power = 1.0)") +  
 theme\_bw()

