Ridgefield to 2018 spatial analyses

Note: not all the code is shown.

use these packages / options

```
library(sp)
library(lctools)
library(gstat)
library(sf)
library(maptiles)
library(prettymapr)
library(fields)
library(viridis)
library(ncf)
library(flextable)
  set_flextable_defaults(font.family = 'sans', theme_fun = "theme_zebra",
      font.size = 11, text.align = "right",padding.left = 1, padding.right = 1,
      padding.top = 2, padding.bottom = 2)
```

Read the data

```
git <- "https://github.com/Ratey-AtUWA/ridgefield-2022/raw/main/"
```

```
rf18a <- read.csv(paste0(git,"rf2018A.csv"), stringsAsFactors = TRUE)
rf_boundary <- read.csv(paste0(git,"rf_boundary.csv"))
```

Making our base map of Ridgefield

```
LongLat <- CRS("+proj=longlat +ellps=WGS84
           +datum=WGS84 +no_defs") # uses Earth ellipsis specs from WGS84 datum
UTM50 <- CRS("+proj=utm +zone=50 +south") # just for Zone 50, S hemisphere!
```

Map extent object

```
extent <- st_as_sf(x = data.frame(x = c(116.94,117.01),
                                  y = c(-32.515,-32.465)),
                   coords = c("x", "y"), crs = LongLat)
st_coordinates(extent)
extent_UTM <- st_transform(extent, crs=UTM50)
st_coordinates(extent_UTM)

##        X       Y
## 1 116.94 -32.515
## 2 117.01 -32.465
##          X       Y
## 1 494364.6 6402477
## 2 500939.8 6408021
```

Getting and plotting the map tile data

```
rftiles <- get_tiles(extent_UTM, provider = "Esri.WorldImagery",
                     crop = TRUE, zoom = 13) # make map object
par(oma=c(3,3,1,1), mar=c(4,4,1.5,1.5), mgp=c(1.4,0.3,0),
    lend=2, ljoin=1, tcl=0.3)
plot(rftiles)
axis(1)
mtext(side=1, line=2, text="Easting (UTM Zone 50, m)",
      font=2, cex=1.2)
axis(2)
mtext(side=2, line=2, text="Northing (UTM Zone 50, m)",
      font=2, cex=1.2)
```

```
with(rf18a, points(Easting, Northing, pch = 3, cex = 0.75, col = "gold"))
# with(rf_border, lines(Easting,Northing, lty = 3, col = "darkgreen"))
polygon(rf_boundary,lty=c("12"),lwd=2,border="#FFFFFFC0")
addnortharrow(pos="topright", border=1, lwd=1, text.col=10,
              padin=c(0.1,0.2), scale=1.2)
addscalebar(plotepsg = 32750, linecol = "white", label.col = "white",
            widthhint = 0.3, htin = 0.15, label.cex = 1.3)
box()
```

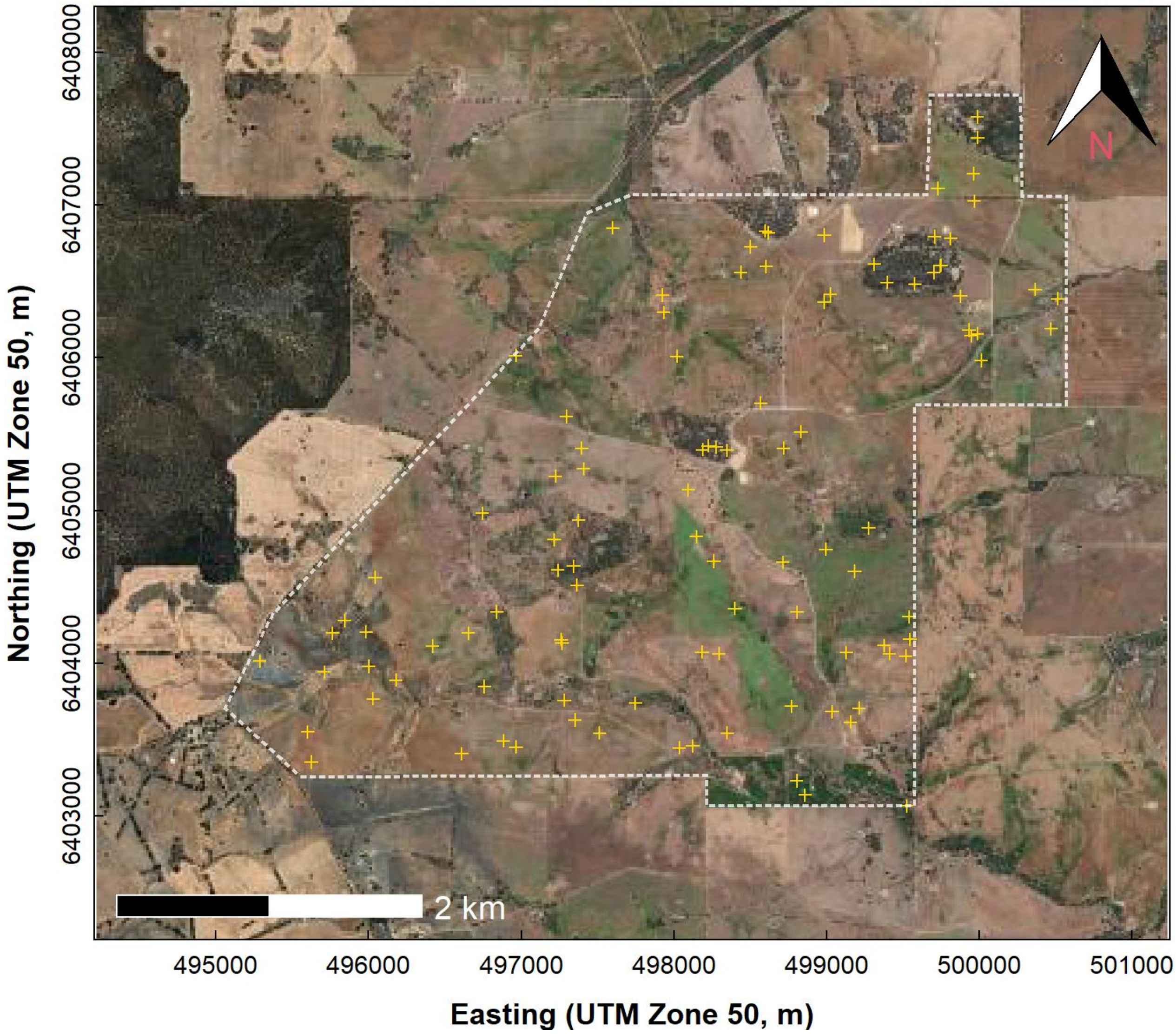

Figure 1: Map of Ridgefield Farm and adjacent area (UTM projection, Zone 50S, EPSG:32750) used subsequently as the base map for spatial analyses. Generated using the maptiles R package, with ESRI WorldTopoMap tiles.

Spatial Autocorrelation

Calculate Global Moran's I

```
var0 <- "C.pct" # choose the variable of interest
# Calculate global Moran's I
data_temp <- na.omit(rf18a[,c("Easting", "Northing", var0)])
Coords <- cbind(data_temp$Easting, data_temp$Northing)
bw <- 8 # bw = bandwidth, = number of nearest point for calculation
mI <- moransI(Coords, bw, data_temp[,3])
cat("Global Moran's I for ",var0,"; from the NGSA (af only), topsoil, -2mm fraction\n", s
ep="");
t(as.matrix(mI[c(2,4:7)]))
rm(list = c("data_temp","Coords","bw","mI","var0")) # remove temporary objects
```

```
## Global Moran's I for C.pct; from the NGSA (af only), topsoil, -2mm fraction
##      Morans.I  z.resampling z.randomization p.value.resampling
## [1,] 0.1184981 2.874977     3.110821        0.004040573
##      p.value.randomization
## [1,] 0.001865678
```

Plot local Moran's I

```
palette("default");palette(c(palette(),"gray92","white","transparent"))
var0 <- "C.pct" # choose the variable of interest
data_temp <- na.omit(rf18a[,c("Easting", "Northing", var0)])
Coords <- cbind(data_temp$Easting, data_temp$Northing)
mI <- moransI(Coords, 8, data_temp[,3]) # log10 minimises skewness
local_moran <- l.moransI(Coords, 8, data_temp[,3], scatter.plot = FALSE)
plotdata <- data.frame(Easting=Coords[,1], Northing=Coords[,2],
                       MoranI=local_moran$Ii, p_value=local_moran$p.value)
pos0 <- subset(plotdata, plotdata$MoranI>0 & plotdata$p_value<=0.05)
neg0 <- subset(plotdata, plotdata$MoranI<0 & plotdata$p_value<=0.05)
#
par(oma=c(3,3,1,1), mar=c(4,4,1.5,1.5), mgp=c(1.4,0.2,0),
    lend=2, ljoin=1, tcl=0.3, lwd = 1)
layout(matrix(c(1,1,1,1,2),nrow = 1))
plot(rftiles)
axis(1)
mtext(side=1, line=1.7, text="Easting (UTM Zone 50, m)",
      font=2, cex=1.2)
axis(2)
mtext(side=2, line=1.7, text="Northing (UTM Zone 50, m)",
      font=2, cex=1.2)
addnortharrow(pos="topright", border=1, lwd=1, text.col=10,
              padin=c(0.1,0.2), scale=1.2)
addscalebar(plotepsg = 32750, linecol = "white", label.col = "white",
            widthhint = 0.3, htin = 0.15, label.cex = 1.3)
box()
points(Coords, pch=3, cex=0.5, col = "lemonchiffon")
with(pos0, symbols(Easting, Northing, circles = 250*sqrt(MoranI*0.02), lwd=2,
                   inches = F, fg = "cyan", bg = "#00808080", add= TRUE))
with(neg0, symbols(Easting, Northing, squares = 250*sqrt(MoranI*-0.04), lwd=2,
                   inches = F, fg = "plum2", bg = "#c0404080", add= TRUE))
plot(c(0,1),c(0,1),type="n",bty="n",axes=F)
legend("top", bty = "o", cex = 1.5, title.cex = 1.5,
       legend = c("Positive I", "Negative I",NA, NA,NA,NA),
       title = paste0(" Local Moran's I for \n",var0,
                      " in rf18a data \n(p \u2264 0.05 only)"),
       pch = c(21, 22,NA,NA,NA), pt.cex = c(3, 2.5,NA,NA,NA,NA),
       col = c("cyan", "plum2"), pt.bg = c("#00808080", "#C0404080"),
       text.col = "white", title.col = "pink", y.intersp = 0.9, bg="#384838")
text(0.5, 0.815, col = 10, cex = 1.2,
     labels = paste("Global Moran's I:\n", signif(as.numeric(mI[2]),3),
                    "\nRandomization p:\n", signif(as.numeric(mI[7]),3)))
```

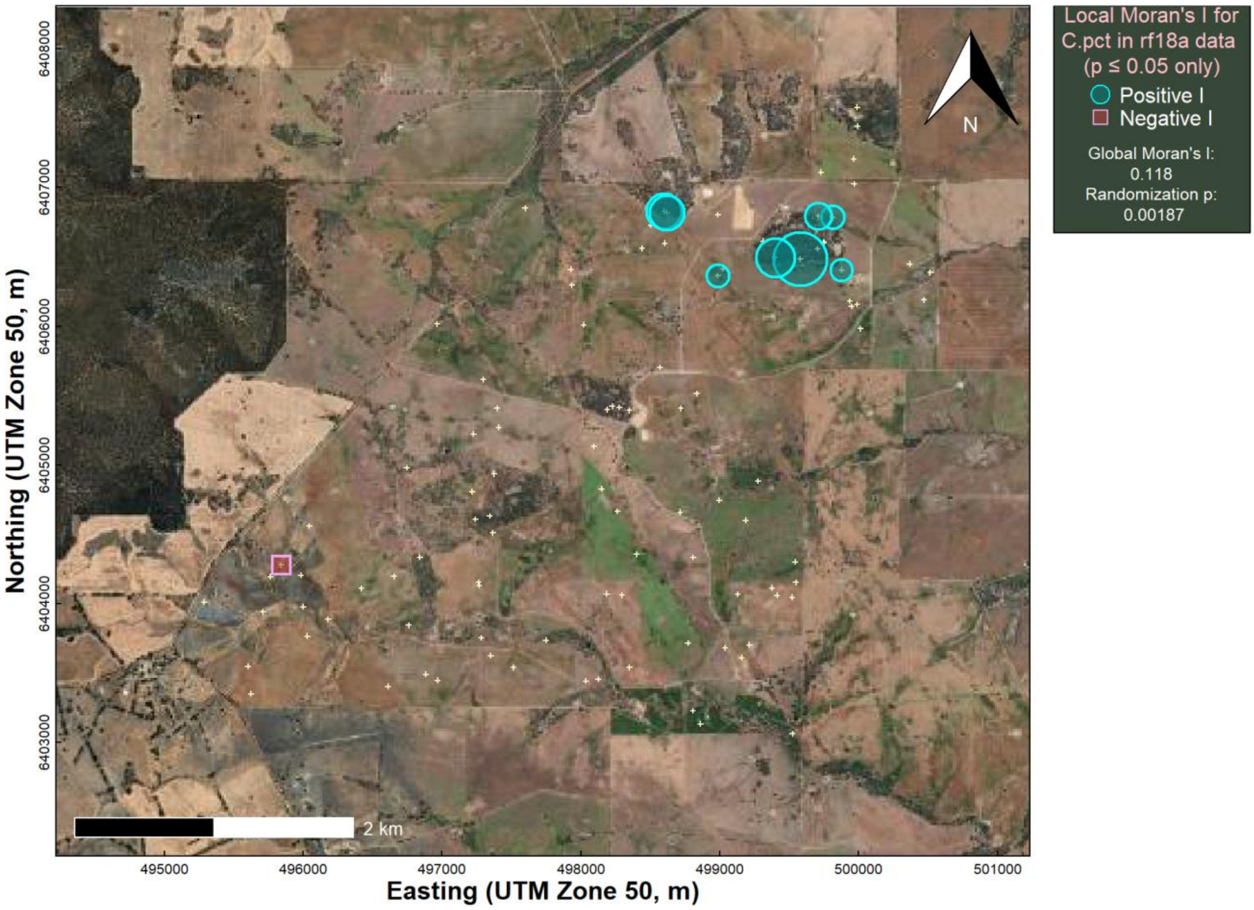

Figure 2: Map of Local Moran's I for TOC concentrations in Ridgefield soils. The Global Moran's I parameter is also shown beneath the legend.

Plot 'LISA'

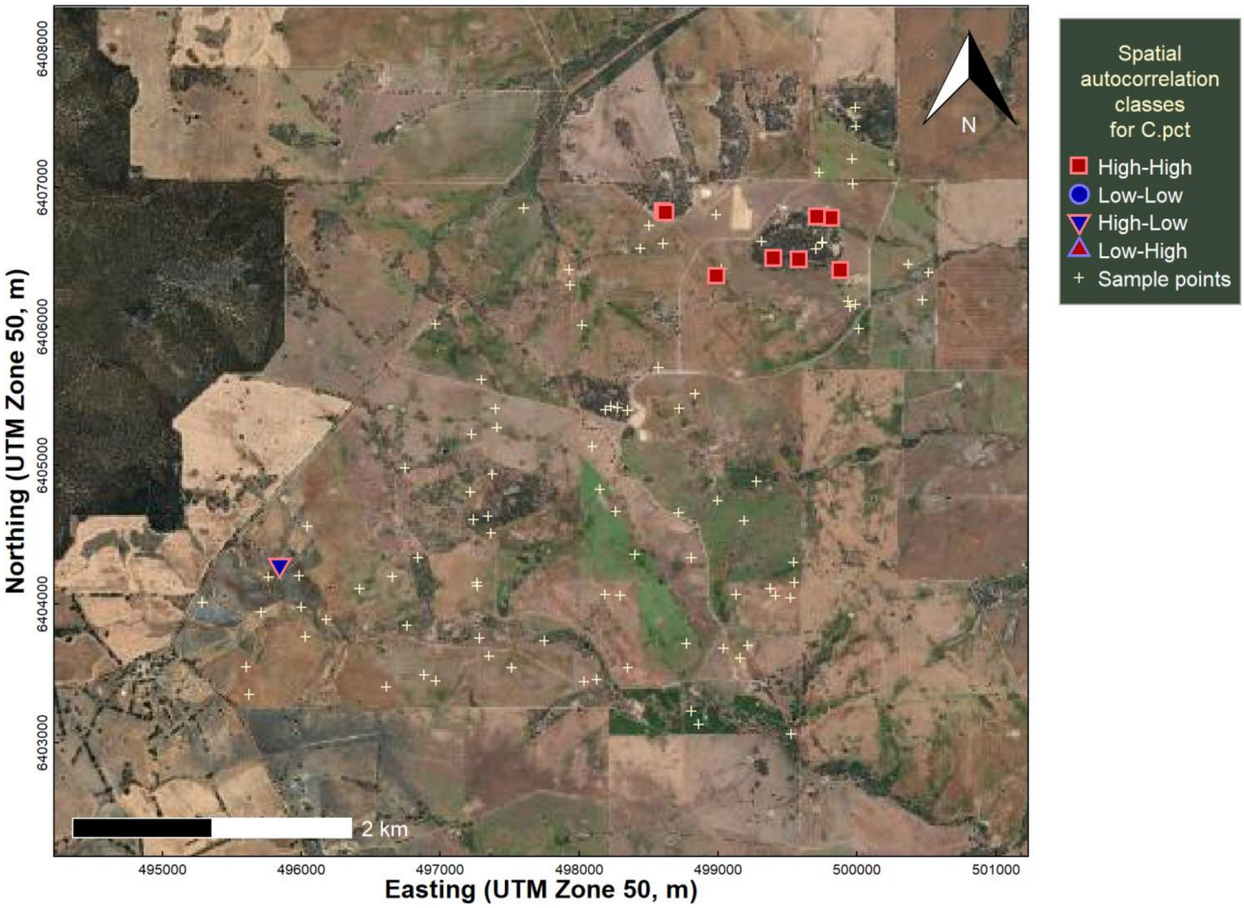

Figure 3: Map of LISA (Local Indicators of Spatial Association) for C.pct in the Ridgefield soil data.

LISA maps for multiple variables

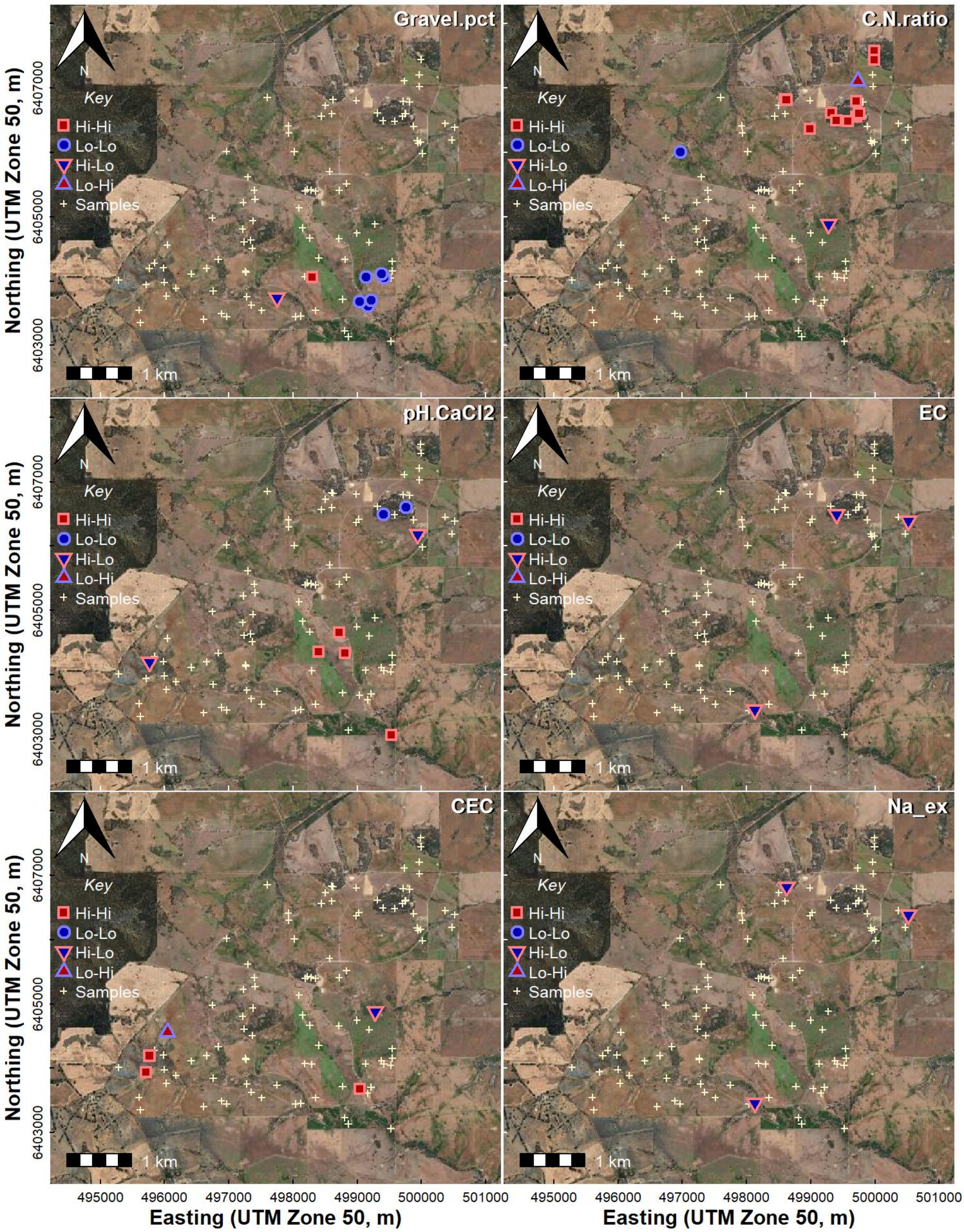

Figure 4: LISA autocorrelation maps for C.pct, Al, Fe, S, As, Cu, Pb, and Zn concentrations in the Ridgefield soil data, 2019-2022.

Using the `gstat` package for geostatistics: variograms, kriging, and visualization

Make `sf` & `SpatialPointsDataFrame` objects from a data frame

In this example (as for Moran's I) we \(log_{10}\)-transform our variable if its distribution is highly positively skewed. (Using the untransformed variable would result in too many apparent upper outliers.)

```
data0 <- na.omit(rf18a[,c("Easting","Northing","C.pct")])
data0[,3] <- log10(data0[,3])
C.pct_sf <- st_as_sf(data0, coords=c("Easting","Northing"), crs = st_crs(32750))
C.pct_sp <- as_Spatial(C.pct_sf)
summary(C.pct_sp)
## Object of class SpatialPointsDataFrame
## Coordinates:
##                 min     max
## coords.x1   495287  500517
## coords.x2 6403063 6407578
## Is projected: TRUE
## proj4string :
## [+proj=utm +zone=50 +south +datum=WGS84 +units=m +no_defs]
## Number of points: 102
## Data attributes:
##      C.pct
##  Min.   :-1.0000
##  1st Qu.: 0.1868
##  Median : 0.3424
##  Mean   : 0.3231
##  3rd Qu.: 0.4878
##  Max.   : 1.2579
```

Plot the spatial object for checking

To quickly check our data, we use the function `bubble()` from the `sp` package to make a *bubble map* of our variable, where the symbol *area* is proportional to the variable value (in this case, soil TOC concentration).

```
bubble(C.pct_sp, col=c("#ff000088", "#0000ff88"), main = "",
       scales = list(draw = TRUE), xlab = "Easting", ylab = "Northing")
```

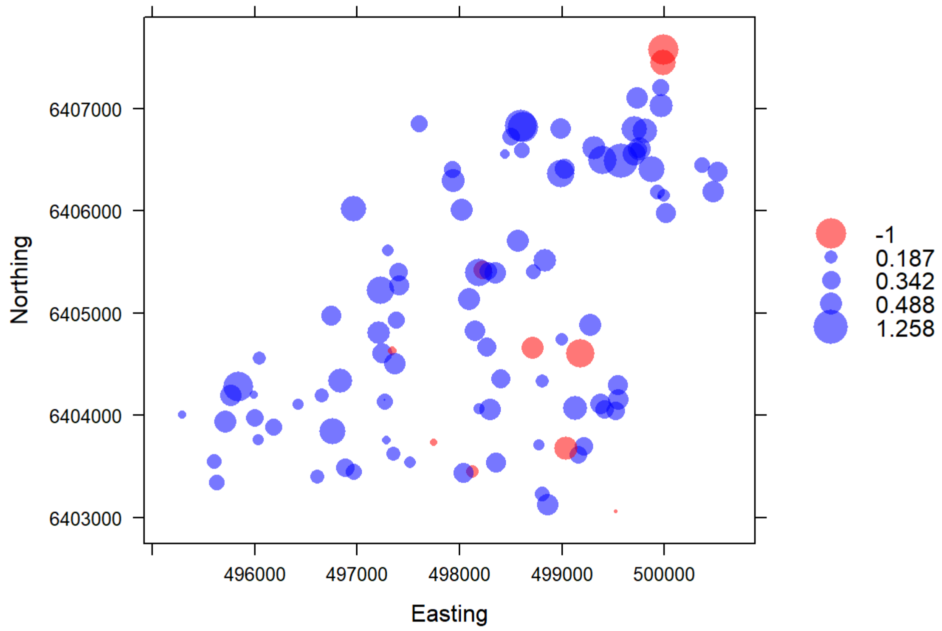

Figure 5: Visualization of spatial point data object for log10-transformed C.pct concentrations in Ridgefield soils 2019-2022.

Plot a map with range-class symbols

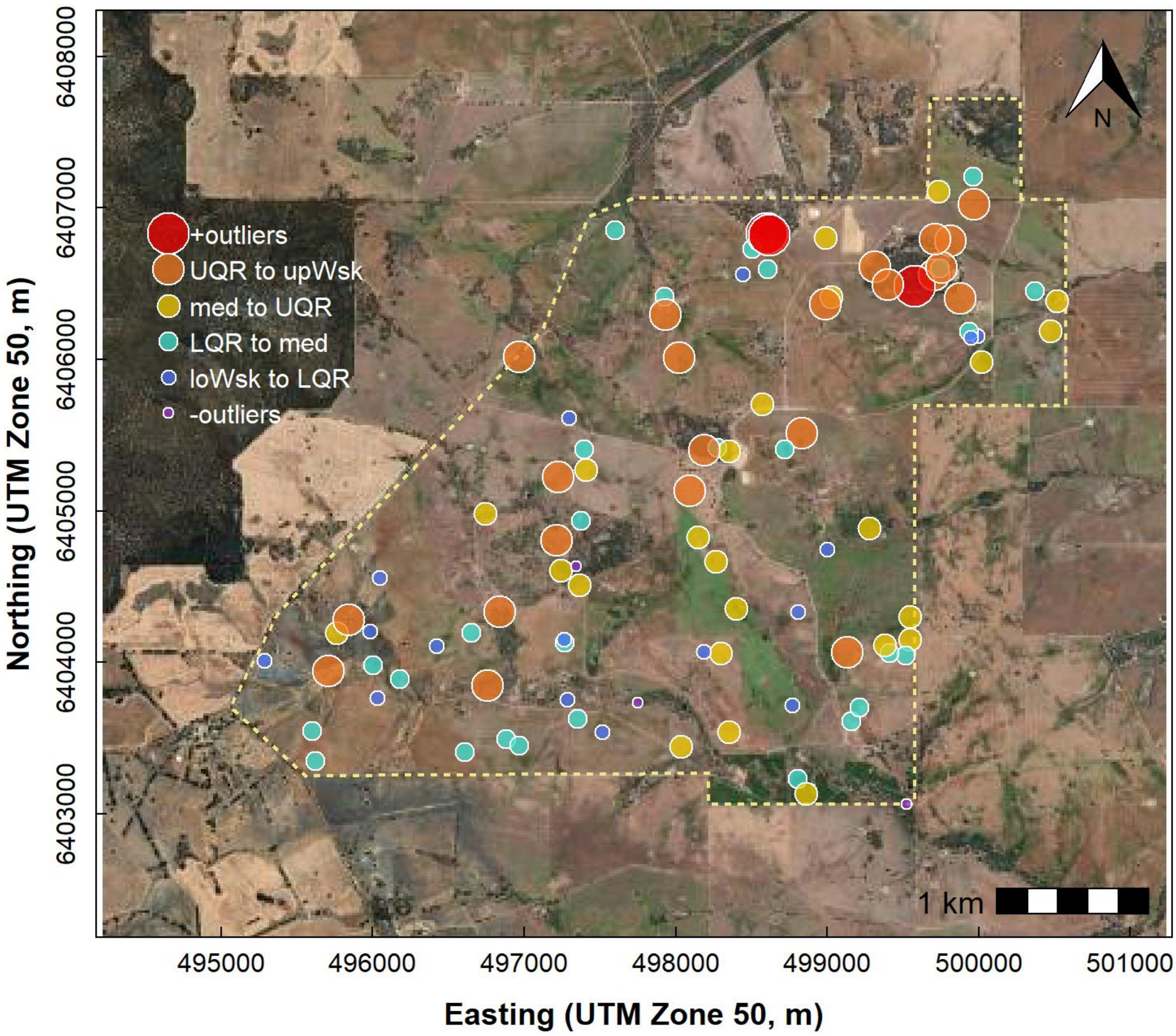

Figure 6: Map of C.pct concentrations expressed as symbols for concentration ranges (UQR is 75th percentile, upWsk is upper whisker, med is median, LQR is 25th percentile, loWsk is lower whisker). Data are from Ridgefield soils 2019-2022.

Table 1: Bins for boxplot-categorised-point map in Figure *6*:

bins	logfrom	logto	from	to
-outliers	-4.000	-0.155	0	0.70
loWsk to LQR	-0.155	0.185	0.70	1.53
LQR to med	0.185	0.342	1.53	2.20
med to UQR	0.342	0.491	2.20	3.10
UQR to upWsk	0.491	0.945	3.10	
+outliers	0.945	Inf	8.81	Inf

Variograms and Kriging

Make a binned simple variogram object

```
data0 <- na.omit(rf18a[,c("Easting","Northing","Gravel.pct")])
data0 <- data0[-which(data0$Gravel.pct<1e-4),] # find and remove zeros
data0[,3] <- log10(data0[,3])
Gravel.pct_sf <- st_as_sf(data0, coords=c("Easting","Northing"), crs = st_crs(32750))
Gravel.pct_sp <- as_Spatial(Gravel.pct_sf)

par(mar = c(3,3,1,1), mgp = c(1.7,0.3,0), font.lab=2, tcl=0.3)

variog_Gravel.pct <- variogram(Gravel.pct~1, Gravel.pct_sf, cressie = TRUE) #
```

```
plot(variog_Gravel.pct$dist, variog_Gravel.pct$gamma, xlab = "Distance (m)",
     pch=19, col="blue2",
     ylab = expression(bold(paste("Semivariance (",gamma,")"))),
     ylim = c(0,max(variog_Gravel.pct$gamma)))
mtext(paste(attributes(variog_Gravel.pct)$what,"with",
            length(variog_Gravel.pct$dist),"bins"), side=1, line=-1.5, adj=0.98)
```

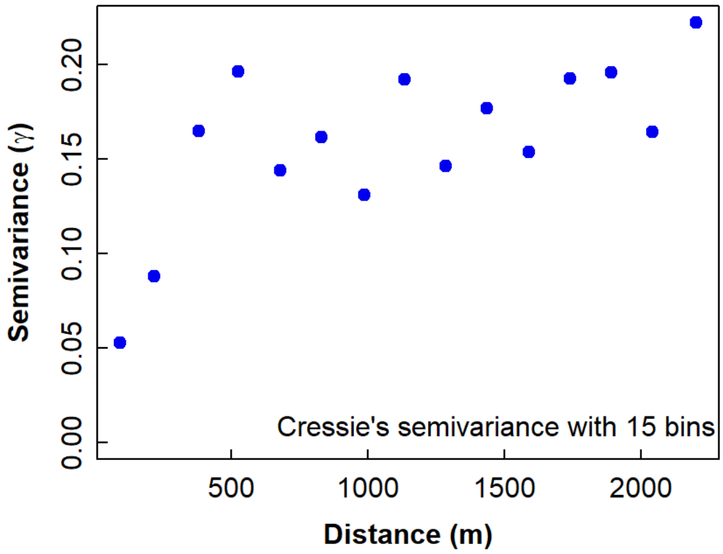

Figure 7: Plot of experimental binned variogram for C.pct in Ridgefield soils 2019-2022.

Table 2: Data for Figure 7: `np` is number of points, `gamma` (γ) is semivariance.

np	dist	gamma
29	94.3	0.053
65	220.6	0.088
93	382.6	0.165
127	527.3	0.196
146	679.9	0.144
171	829.9	0.162
184	989.7	0.131
187	1,136.4	0.192
184	1,287.5	0.146
191	1,438.5	0.177
208	1,590.1	0.154
184	1,740.6	0.192
251	1,893.7	0.196
215	2,042.4	0.164
202	2,202.6	0.222

Fit a variogram model using weighted least squares

```
vmodel_Gravel.pct <- fit.variogram(variog_Gravel.pct,
                              model = vgm(psill = NA, model = "Exp",
                                          nugget = 0.2,
                                          cutoff = max(variog_Gravel.pct$dist)))
vmodel_Gravel.pct
##   model     psill    range
## 1   Nug 0.0000000   0.0000
## 2   Exp 0.1784494 259.0698
```

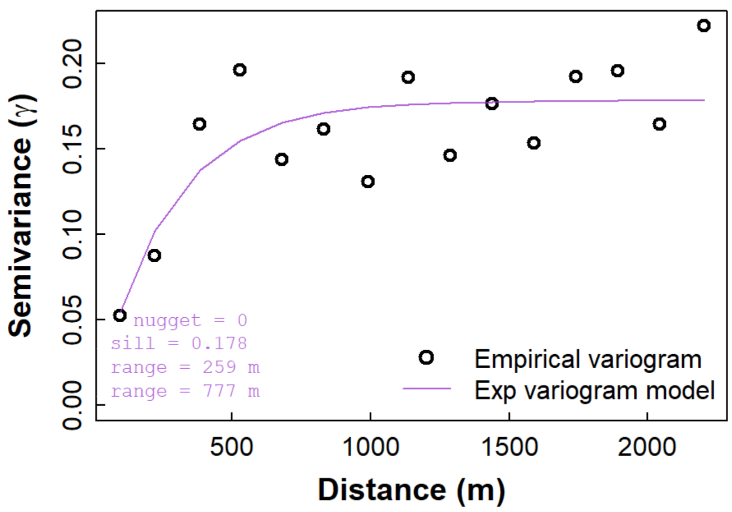

Figure 8: Plot of experimental binned variogram, and exponential variogram model, for Gravel.pct in Ridgefield soils 2019-2022.

Perform kriging interpolation

first make a grid mask

```
grid0 <- expand.grid(seq(round(min(Gravel.pct_sp@coords[,1]),1),
                            round(max(Gravel.pct_sp@coords[,1]),1), 10),
                        seq(round(min(Gravel.pct_sp@coords[,2]),1),
                            round(max(Gravel.pct_sp@coords[,2]),1), 10))
afgrid <- SpatialPoints(grid0, proj4string = UTM50)

irregpoly <- Polygon(rf_boundary[,c(1,2)], hole=F)
irregPolys = Polygons(list(irregpoly),1)
gridMask = SpatialPolygons(list(irregPolys),
                           proj4string = UTM50)
inOrOut <- as.vector(over(afgrid, gridMask))
afgrid <- afgrid[which(inOrOut>0)]
cat("Prediction grid:\n"); summary(afgrid)
## Prediction grid:
## Object of class SpatialPoints
## Coordinates:
##          min     max
## Var1  495287  500517
## Var2 6403073 6407433
## Is projected: TRUE
## proj4string :
## [+proj=utm +zone=50 +south +datum=WGS84 +units=m +no_defs]
## Number of points: 145410
```

Krige to grid

```
kriged_Gravel.pct <- krige(formula = Gravel.pct~1, locations = Gravel.pct_sp,
                      newdata = afgrid, model = vmodel_Gravel.pct)
summary(kriged_Gravel.pct)
cat("______________________________________________________________________\n\n")
idw_Gravel.pct <- idw(formula = Gravel.pct~1, locations = Gravel.pct_sp,
                      newdata = afgrid, idp = 2)
idw_Gravel.pct@data$var1.var <- NULL
summary(idw_Gravel.pct)
## [using ordinary kriging]
## Object of class SpatialPointsDataFrame
## Coordinates:
##          min     max
## Var1  495287  500517
## Var2 6403073 6407433
## Is projected: TRUE
## proj4string :
```

```
## [+proj=utm +zone=50 +south +datum=WGS84 +units=m +no_defs]
## Number of points: 145410
## Data attributes:
##    var1.pred          var1.var
##  Min.   :-1.0000   Min.   :0.00000
##  1st Qu.: 0.8646   1st Qu.:0.09822
##  Median : 0.9801   Median :0.12744
##  Mean   : 0.9465   Mean   :0.12172
##  3rd Qu.: 1.1171   3rd Qu.:0.15118
##  Max.   : 1.9536   Max.   :0.17922
## ______________________________________________________________________
##
## [inverse distance weighted interpolation]
## Object of class SpatialPointsDataFrame
## Coordinates:
##          min     max
## Var1  495287  500517
## Var2 6403073 6407433
## Is projected: TRUE
## proj4string :
## [+proj=utm +zone=50 +south +datum=WGS84 +units=m +no_defs]
## Number of points: 145410
## Data attributes:
##    var1.pred
##  Min.   :-1.0000
##  1st Qu.: 0.9154
##  Median : 1.0191
##  Mean   : 0.9660
##  3rd Qu.: 1.1053
##  Max.   : 1.9698
```

Simple plot of kriging output

We can then us the `spplot()` function from `sp` to visualise the kriging predictions and variance, but without a background map.

```
spplot(kriged_Gravel.pct, colorkey=T, pch=15, # with kriging variances
    names.attr = c("log10[Gravel.pct] kriging predictions",
                   "log10[Gravel.pct] kriging variance"),
    scales = list(draw = TRUE), col.regions = viridis::mako(100),
    xlab="Longitude", ylab="Latitude")
```

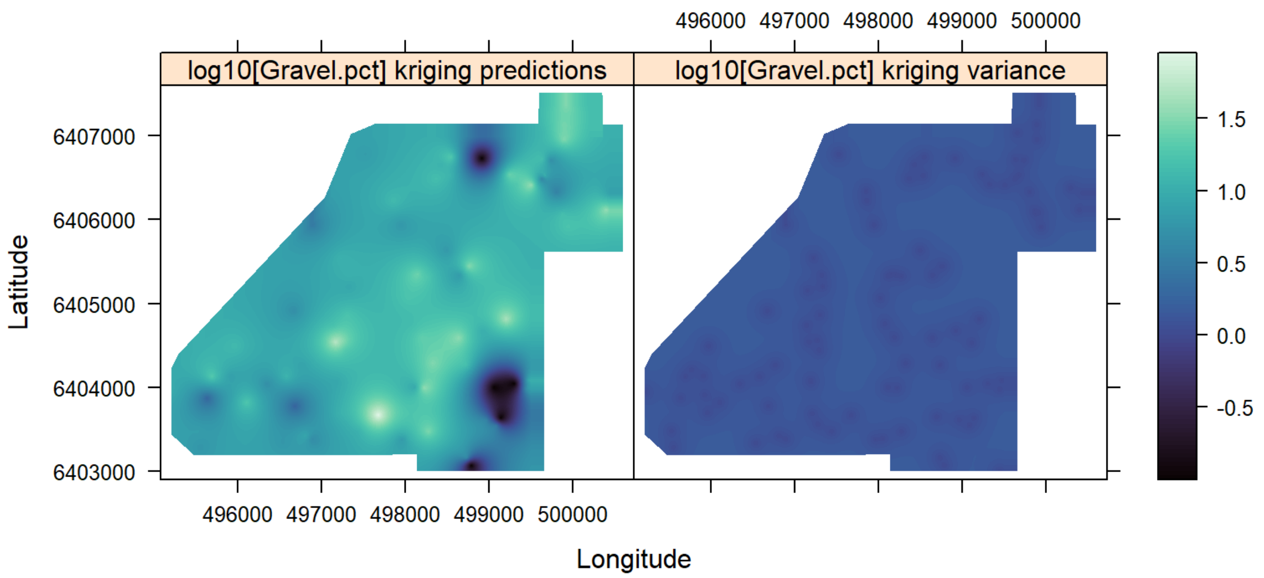

Figure 9: Plots of kriging predictions and variance for log-transformed Gravel.pct in Ridgefield soil.

Plot a nicer map with overlay of the kriging predictions & kriging variance

```
par(mfrow=c(2,1), oma=c(4,4,1.5,1.5), mgp=c(1.4,0.3,0), lend=2, ljoin=1, tcl=0.3)
plot(rftiles)
axis(1, mgp=c(2, 0.3, 0), labels=F)
# mtext(side=1, line=1.5, text="Easting (UTM Zone 50, m)",
#       font=2, cex=1.2)
axis(2, mgp=c(2, 0.3, 0), cex.axis = 1)
mtext(side=2, line=1.5, text="Northing (UTM Zone 50, m)",
      font=2, cex=1.2)
axtx <- c(1,2,3,5,10,20,30,50,100,200,300,500,1000)
colgrad <- gsub("FF","98", viridis::mako(128)[round(seq(1, 128, l=6),0)])
rect(494500, 6402250, 501000, 6403000, col="#FFFFFF80", border=NA)
quilt.plot(afgrid@coords[,1], afgrid@coords[,2],kriged_Gravel.pct@data[,1],
           add = T, horizontal=T,
           col = colorRampPalette(colgrad, alpha = TRUE)(64),
           legend.lab = expression(bold("Kriging prediction of Gravel.pct (mg/kg)")),
           legend.mar = 6, legend.cex = 1, legend.line = 1.2, text.col=10,
           axis.args = list(at=log10(axtx), labels=axtx, mgp=c(1.2,0.1,0)))
addnortharrow(border=1, lwd=1, text.col=1, padin=c(0.1,0.1), scale = 0.9)
addscalebar(plotepsg = 32750, pos = "topleft")
points(rf18a[,c("Easting","Northing")], pch=16, cex=0.1, col = "gold")
box() ; mtext("(a)",3,-1.5, cex=1.5,col=10)

plot(rftiles)
axis(1, mgp=c(2, 0.3, 0), cex.axis = 1)
mtext(side=1, line=1.5, text="Easting (UTM Zone 50, m)",
      font=2, cex=1.2)
axis(2, mgp=c(2, 0.3, 0), cex.axis = 1)
mtext(side=2, line=1.5, text="Northing (UTM Zone 50, m)",
      font=2, cex=1.2)
axtx <- seq(0,0.14,0.02)
colgrad <- gsub("FF","80", viridis::turbo(128)[round(seq(1, 128, l=6),0)])
rect(494500, 6402250, 501000, 6403000, col="#FFFFFF80", border=NA)
quilt.plot(afgrid@coords[,1], afgrid@coords[,2],kriged_Gravel.pct@data[,2],
           add = T, horizontal=T,
           col = colorRampPalette(colgrad, alpha = TRUE)(64),
           legend.lab = expression(bold("Kriging variance in Gravel.pct (mg/kg)")),
           legend.mar = 6, legend.cex = 1, legend.line = 1.2, legend.col="white",
           axis.args = list(at=axtx, labels=axtx, mgp=c(1.2,0.1,0)))
addnortharrow(border=1, lwd=1, text.col=1, padin=c(0.1,0.1), scale = 0.9)
addscalebar(plotepsg = 32750, pos = "topleft")
points(rf18a[,c("Easting","Northing")], pch=16, cex=0.1, col="orchid")
box() ; mtext("(b)",3,-1.5, cex=1.5,col=10)
```

Maps are in Figure 10 on next page.

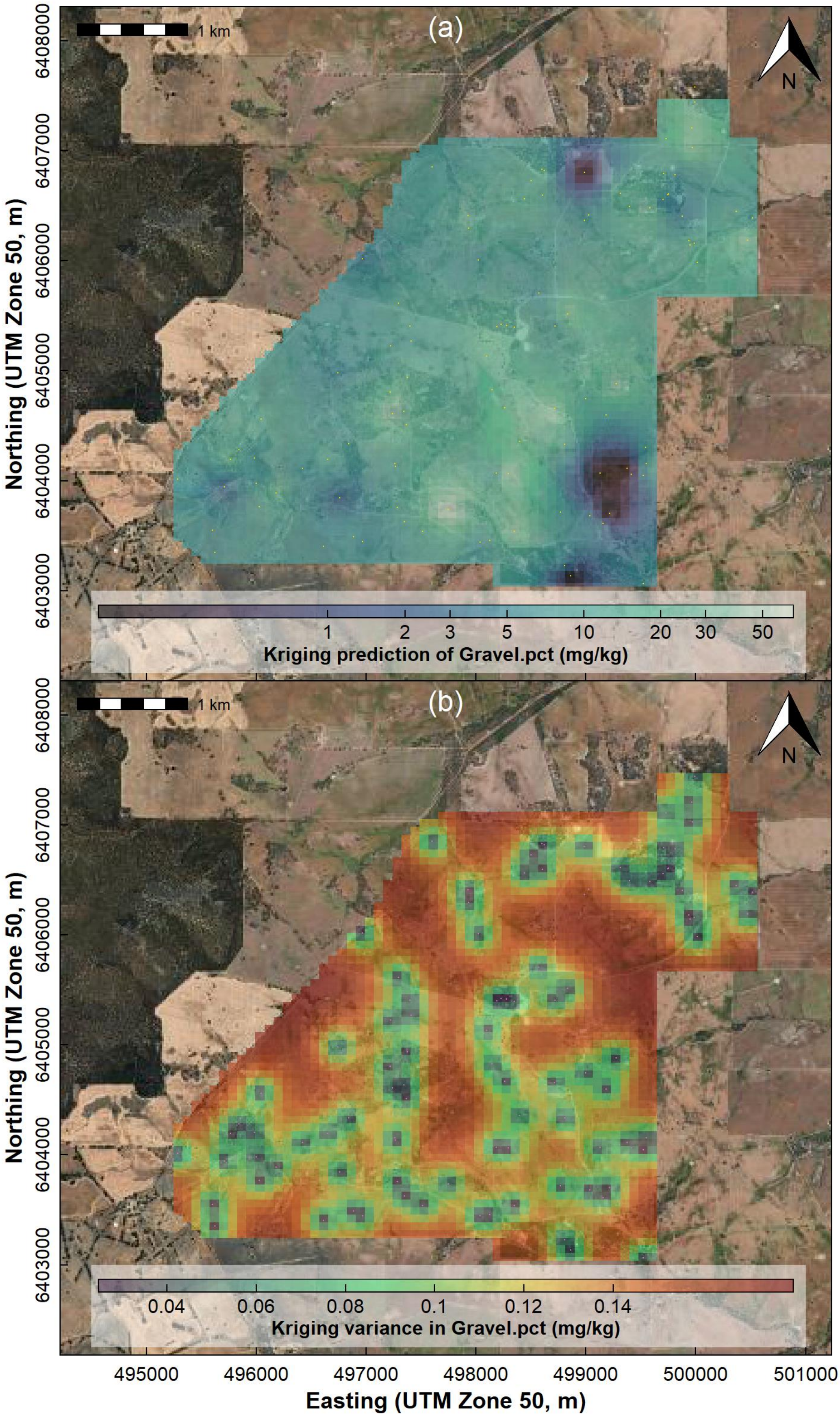

Figure 10: Map showing (a) predictions and (b) variance for kriging model of Gravel.pct in Ridgefield soils. Sample points are tiny dots.