Aufgabe 06

Gruppe 01

4 6 2020

Laden Sie den Workspace yingtan_20_ueb3.Rdata sowie das Paket gstat und überführen Sie das Objekt ljz in ein SpatialPointsDataFrame. Reproduzieren Sie ihr Variogrammmodell aus der vorangegangenen Übung.

```
library(tidyverse)
```

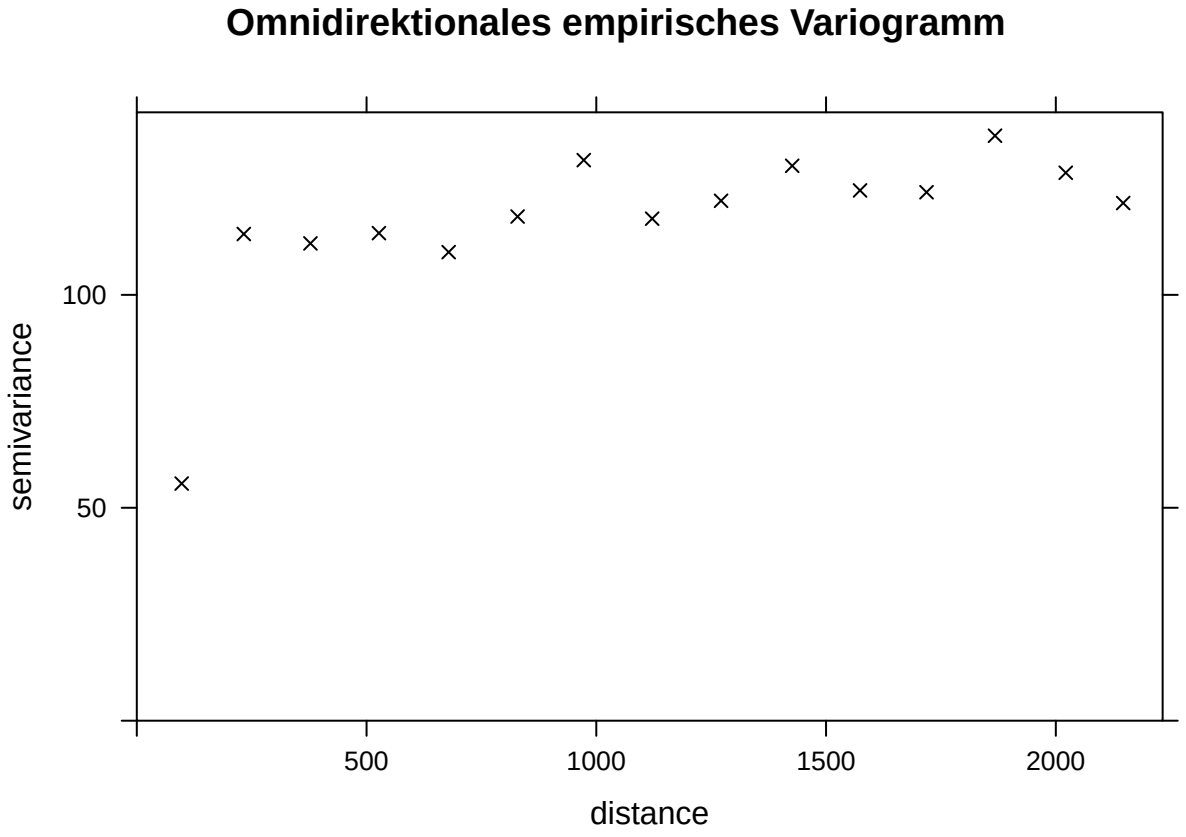
```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

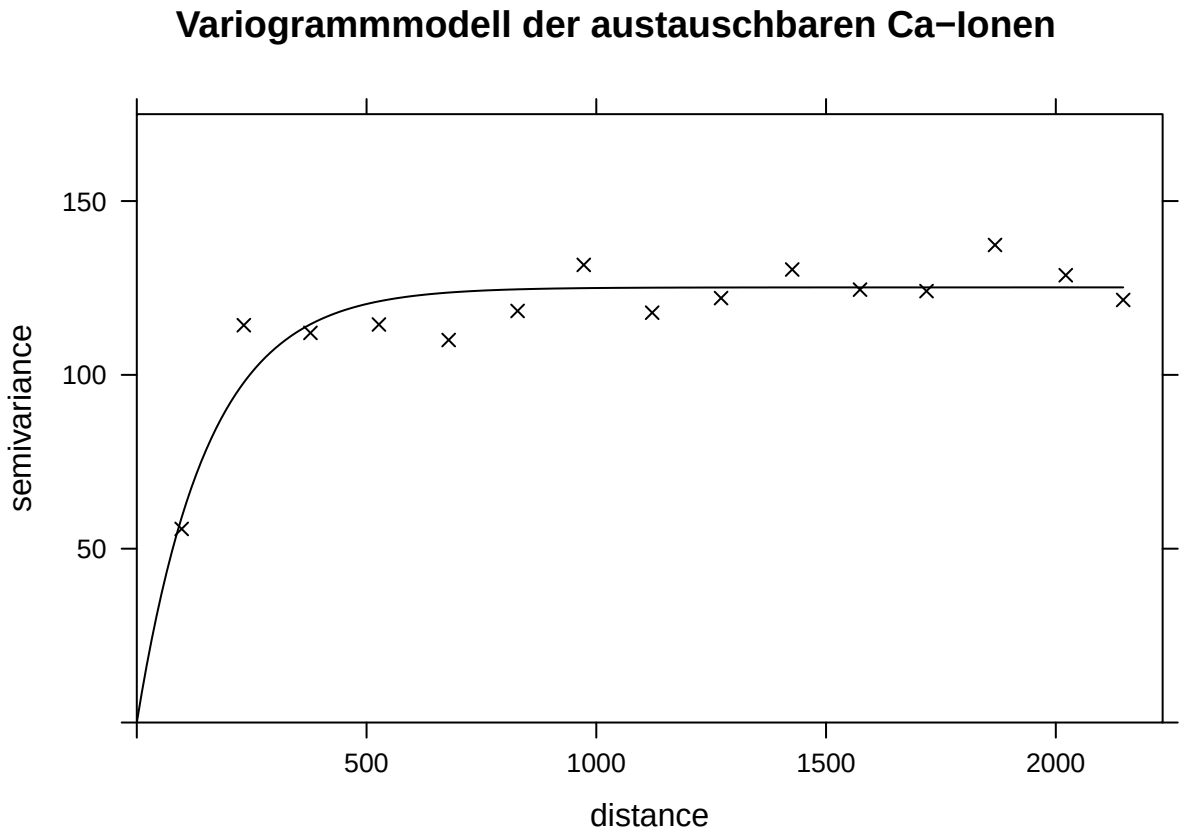
```
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
load("data/yingtan_20_ueb3.RData")

##SpatialPointsDataFrame##
library(sp)
#Spalten mit den Koordinaten selektieren
spatialCoords <- dplyr::select(ljz,
                               long = EAST,
                               lat = NORTH)

#Koordinatensystem definieren (s. txt-Datei Datenbeschreibung)
coordRefSys <- CRS("+proj=utm +zone=50 +ellps=WGS84 +datum=WGS84")

#SpatialPointsDataFrame erstellen
SPDFljz <- SpatialPointsDataFrame(spatialCoords,
                                  ljz,
                                  proj4string = coordRefSys)

##Reproduktion des Variogrammmodells##
#omnidirektionales empirisches Variogramm
library(gstat)
Ca <- SPDFljz@data$Ca_exch

vario_omni_Ca <- variogram(Ca ~ EAST + NORTH,
                           data = SPDFljz,
                           cutoff = 2202,
                           width = 150)

plot(vario_omni_Ca,
     main = "Omnidirektionales empirisches Variogramm",
```

```
     pch = 4,
     col = "black")
```




```
#Modell zum Variogramm
vario_omni_Ca_fit <- fit.variogram(vario_omni_Ca,
                                   vgm(model = "Exp"))

plot(vario_omni_Ca,
     model = vario_omni_Ca_fit,
     cutoff = 2202,
     ylim = c(0, 175),
     pch = 4,
     col = "black",
     main = "Variogrammmodell der austauschbaren Ca-Ionen")
```



Aufgabe 13 Ordinary Kriging

Ordinary Kriging (OK) ist das am häufigsten verwendete Kriging-Verfahren zur Interpolation punktueller Daten in die Fläche. Vom Prinzip her eine spezielle lineare Regressionstechnik, berücksichtigt es räumliche Strukturen durch die Integration der Variogramme in den Berechnungsprozess.

a) Erstellen Sie ein reguläres Raster für die räumliche Interpolation. Entscheiden Sie sich für eine geeignete Pixelgröße und begründen Sie ihre Wahl. Erweitern Sie die bounding box des Objekts ljz in x-Richtung jeweils um 180m und nach Norden und Süden um je 215m, damit auch das komplette, 10.5 km² große, Untersuchungsgebiet erfasst wird. Um ihr Spatial- Grid zu erzeugen, nutzen Sie die Funktion GridTopology. (3 Punkte)

HENGL, T. (2006): Finding the right pixel size. In: Computers & Geosciences, 32: 1283-1298. (Abschnitt 2.5: Grid resolution and point samples)

```
#Anpassung der bounding box
bb <- bbox(SPDFljz)
bb[1,1] <- bb[1,1]-180
bb[1,2] <- bb[1,2]+180
bb[2,1] <- bb[2,1]-215
bb[2,2] <- bb[2,2]+215
bb
```

```
##          min     max
## long  490261  493771
## lat  3121075 3125845
```

```
library(raster)
```

```
##
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(rgdal)
```

```
## rgdal: version: 1.5-8, (SVN revision 990)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.0.4, released 2020/01/28
## Path to GDAL shared files: C:/Users/helen/Documents/R/win-library/4.0/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ_VERSION: 631]
## Path to PROJ shared files: C:/Users/helen/Documents/R/win-library/4.0/rgdal/proj
## Linking to sp version:1.4-2
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading rgdal.
```

```
#Pixelsize nach Hengl 2006
sqrt(10500000/335) * 0.0791
```

```
## [1] 14.00389
```

```
#Spatial Grid
GT <- GridTopology(cellcentre.offset = c(bb[1,1], bb[2,1]),
             cellsize = c(14, 14),
             cells.dim = c(250, 340)) #Ergibt sich gerundet aus Maße der bbox durch 14

targetSPG <- SpatialGrid(grid = GT,
                         proj4string = coordRefSys)
```

b) Wenden Sie die Methode krige entsprechend des Ordinary Kriging an und interpolieren Sie die Konzentration der Ca-Ionen auf die Rasterzellen des in Aufgabe a) erzeugten Objekts. Benutzen Sie ihr Variogrammmodell aus der vorangegangenen Übung und notieren Sie ihre R-Syntax im Protokoll. (1 Punkt)

```
#Ordinary Kriging
interpol_Ca <- gstat::krige(Ca ~ 1,
                            SPDFljz, targetSPG,
                            model = vario_omni_Ca_fit)
```

```
## [using ordinary kriging]
```

```
#summary(interpol_Ca)
#plot(raster(interpol_Ca))
```

c) Welche zwei Daten-Attribute produziert die krige-Funktion und wofür stehen sie? (1 Punkt)

Die krige-Funktion produziert zum einen var1.pred und zum anderen var1.var. var1.pred beinhaltet die vorhergesagten Werte für die austauschbaren Ca-Ionen. var1.var beinhaltet die die Varianzen für die berechneten Werte.

d) Ermitteln Sie diejenigen Pixel in dem Vorhersagegrid der Zielgröße, für die die errechnete Kriging-Varianz den Wert der Gesamt-Varianz der Zielgröße überschreitet und definieren Sie diese als NoData. (1 Punkt)

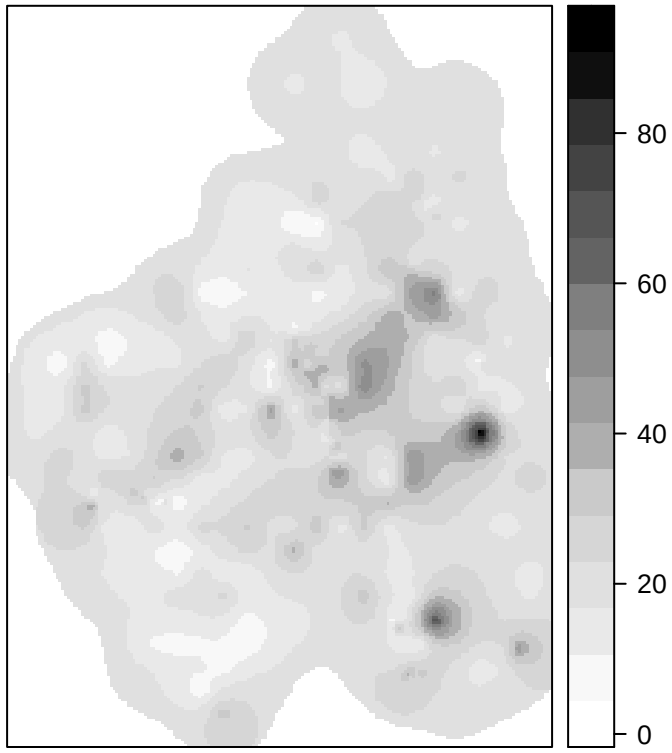
```
var(Ca)
```

```
## [1] 123.8529
```

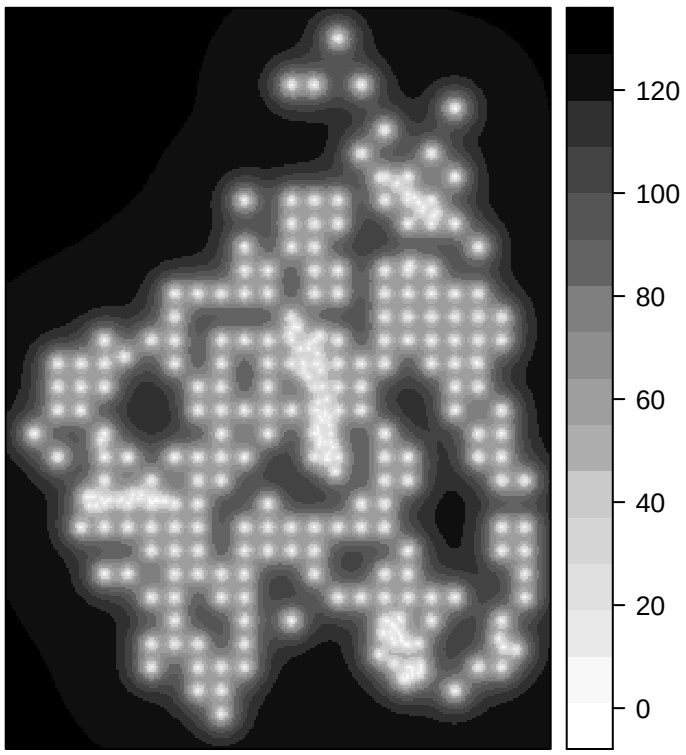
```
#var1.var ist max. 127.13

NoData <- interpol_Ca
NoData[NoData$var1.var > 123.85] <- NA
#NoData
```

e) Plotten Sie sowohl die vorhergesagten Ca-Ionenkonzentrationen als auch die zugehörigen Kriging-Varianzen mit Hilfe der Methode spplot. Maskieren Sie in der Abbildung der Vorhersagewerte die in d) ausgeschlossenen Pixel. Beschriften Sie vernünftig und verändern Sie die Farbskalen beider Graphiken, sodass ein gefälliger Druck in schwarz/weiß möglich ist. (3 Punkte) Hinweis: Mit Hilfe des Befehls ??Palettes erfragen Sie diverse Methoden verschiedener Pakete zur Erstellung von Farbpaletten. Insbesondere das Paket „RColorBrewer“ könnte dabei in Verbindung mit der Website http://colorbrewer2.org/ von Interesse für Sie sein.

```
library(RColorBrewer)
mycolor <- colorRampPalette(brewer.pal(n=9, name = "Greys"))(20)

RNoData <- raster(NoData)
Rinterpol_Ca <- raster(interpol_Ca)

mRinterpol_Ca <- mask(Rinterpol_Ca, RNoData)

#Darstellung
library(lattice)
mylayout <- list(col.pixels = mycolor)
spplot(mRinterpol_Ca, "var1.pred",
       main = "Vorhergesagte Ca-Ionenkonzentration",
       col.regions=mycolor)
```




```
spplot(interpol_Ca, "var1.var",
       main = "(Co)varianzen der Ca-Ionenkonzentration",
       col.regions=mycolor)
```



f) Interpretieren Sie kurz und knapp ihren Kriging-Varianz-Plot. Wieso ist die Kriging-Varianz als internes Gütemaß der Interpolation nur bedingt aussagekräftig? (1 Punkt)

Die Aussagekraft hängt stark von der Pixelgröße und der Beprobungsdichte ab. Je dichter beprobt wurde, desto kleiner kann die Pixelgröße gewählt werden und desto genauer wird die Interpolation. Da es aber meist mehr Einflussfaktoren für einen bestimmten Messwert (hier die Konzentration von austauschbaren Ca-Ionen) als die räumliche Nähe gibt, ist die Interpolation auf Basis der Distanz nur eine Annäherung bzw. eine grobe Schätzung. In diesem Beispiel hat bereits das Variogramm gezeigt, dass nur Orte, die sehr nah beieinander liegen, ähnliche Werte/eine geringe Varianz der Werte aufweisen. Die Abbildung der Varianzen zeigt, dass Pixel höherer Entfernung zum tatsächlich beprobten Standort eine höhere Varianz aufweisen. Damit ist die Aussage über einen Wert für die austauschbaren Ca-Ionen als “sehr unsicher” zu bewerten und eine Interpolation allein nicht aussagekräftig genug.