

Aufgabenzettel 06

Gruppe 01

09.06.2020

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.

```
library(tidyverse)

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

## v ggplot2 3.3.1      v purrr   0.3.4
## v tibble  3.0.1      v dplyr   1.0.0
## v tidyr   1.1.0      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()

library(gstat)
library(sp)

SPDF <- ljz
coordinates(SPDF) <- ~ EAST+NORTH
proj4string(SPDF) <- CRS("+init=epsg:32650")

cloud <- variogram(Ca_exch~1,
  SPDF,
  #cutoff= 2202,
  #width= 1000,
  cloud = TRUE
)

var <- variogram(Ca_exch~1,
  SPDF,
  cutoff=2202, # ca. 50% der Max. Distanz
  width= 100
)

m <- vgm(#psill = 110,
  model = "Exp",
  #model = c("Nug", "Exp", "Log", "Gau"),
```

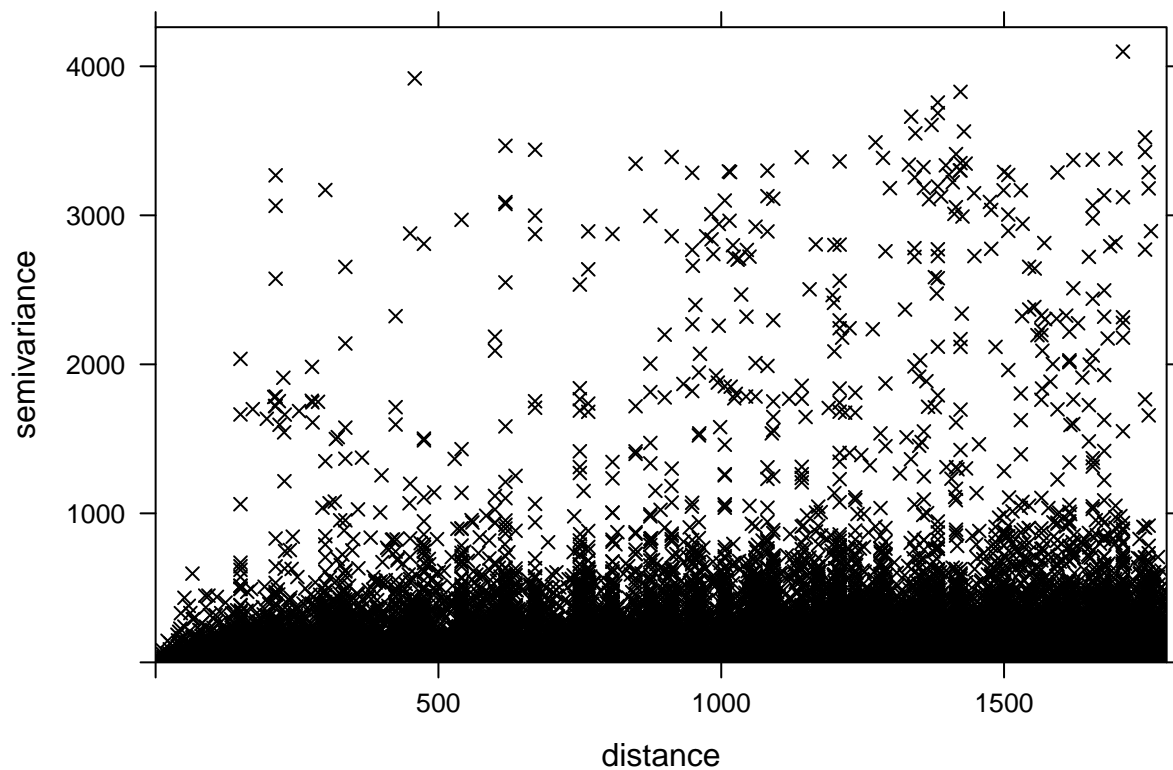
```

#range = 300,
#kappa= 2,
#nugget = 2,
cutoff= 2202)

v_fit <- fit.variogram(object = var,
  model = m,
  #fit.sills = TRUE,
  #fit.ranges = TRUE,
  fit.method = 7, #vgl. Gstat user's manual,
                  #p. 42, tab. 4.2,
                  #<http://www.gstat.org/gstat.pdf>
                  # 7 ordinary least squares
  #fit.kappa = TRUE,
  )

plot(cloud, pch=4, col="black")

```



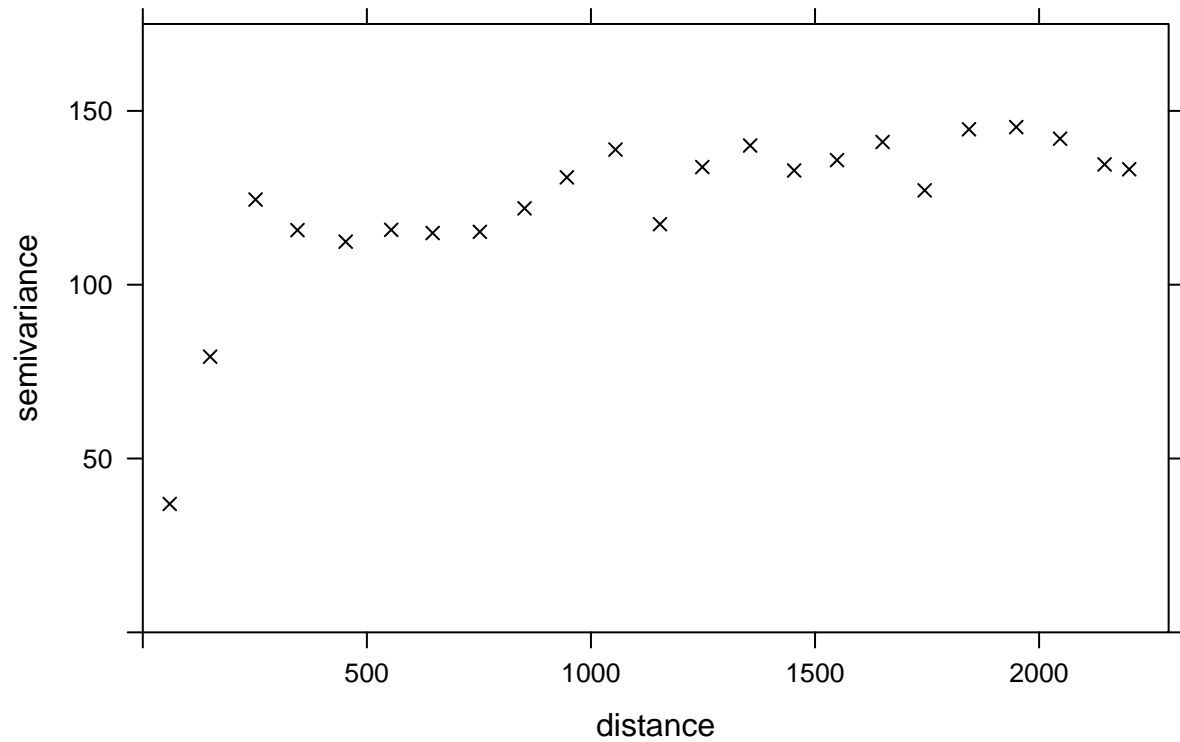
```

plot(var,
  main= "Omnidirektionales empirisches Variogramm",
  pch=4,
  col="black"
  ,omnidirectional=TRUE,
  ylim=c(0,175)

```

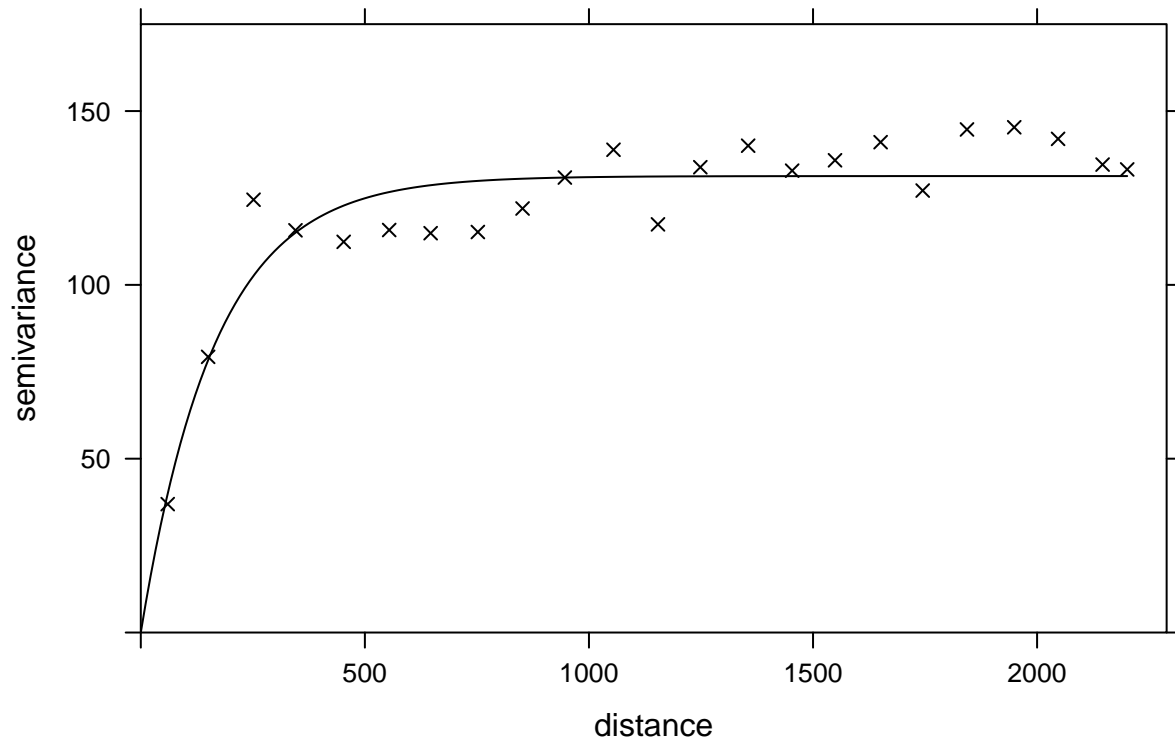
)

Omnidirektionales empirisches Variogramm



```
plot(var,
      #m1,
      v_fit,
      cutoff = 2202,
      ylim=c(0,175),
      pch=4,
      col="black",
      main="Variogramm Modell",
      #xlim=c(0, 2500),
      )
```

Variogramm Modell



13 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 ljk, 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 SpatialGrid zu erzeugen, nutzen Sie die Funktion GridTopology.

Da die Beprobungspunkte aufgrund der Catenen nicht streng nach einem Raster erfolgt ist empfiehlt sich nach Hengl (2006) eine Pixelgröße p von

$$p = 0.25 * \sqrt{\frac{A}{N}}$$

wobei A gleich der Fläche des Untersuchungsgebietes in m² und N gleich der Gesamtzahl an Beobachtungen ist.

```
print(0.25*sqrt((10.5e+6)/length(SPDF)))
```

```
## [1] 44.26009
```

```
print(0.0791*sqrt((10.5e+6)/length(SPDF)))
```

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

```
#Wilke, S. (2020): A practical guide to geospatial interpolation with R.  
#<https://swilke-geoscience.net/post/spatial\_interpolation/>
```

```
#Von Punkten zu Pixeln zu Rastern  
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/Pierre/Documents/R/win-library/3.6/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/Pierre/Documents/R/win-library/3.6/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.

##?raster()

#SpatialPointsDataFrame zu Raster
#e <- extend()
r <- raster(SPDF)

## Warning in proj4string(x): CRS object has comment, which is lost in output

#Grid definieren
g <- GridTopology(cellcentre.offset = c(490441, 3121290), #SPDF@bbox[,1]
  cellsize = c(14,14), #c(14, 14)
  cells.dim = c(70, 97)
)

PixDF <- SpatialGrid(#points = ,
  #data = ,
  proj4string = CRS("+init=epsg:32650"),
  grid = g
)

#PixDF <- rasterize(SPDF, g, PixDF)

# e <- extent(SPDF[,1:2])
# r <- raster(e,resolution=45)
# x <- rasterize(SPDF[,1:2], r, )

##?bbox()

```

```

#?attr()

# SPDF
# b <- bbox(SPDF)
# b
# b[1,1] <- b[1,1]-180;
# b[1,2] <- b[1,2]+180;
# b[2,1] <- b[2,1]-215;
# b[2,2] <- b[2,2]+215;

## Option 1 ?
# bbox <- c(
#   "xmin" = min(pts_NH4$X),
#   "ymin" = min(pts_NH4$Y),
#   "xmax" = max(pts_NH4$X),
#   "ymax" = max(pts_NH4$Y)
# )
#
# grd_template <- expand.grid(
#   X = seq(from = bbox["xmin"], to = bbox["xmax"], by = 20),
#   Y = seq(from = bbox["ymin"], to = bbox["ymax"], by = 20) # 20 m resolution
# )

## Option 2
# fran = franconia
# new_bb = c(8, 48, 12, 50)
# names(new_bb) = c("xmin", "ymin", "xmax", "ymax")
# attr(new_bb, "class") = "bbox"
#
# attr(st_geometry(fran), "bbox") = new_bb

```

13 b)

Wenden Sie die Methode krig entsprechend des Ordinary Kriging an und interpolieren Sie die Konzentration der Ca-Ionen auf die Rasterzellen des in Aufgabe 13 a) erzeugten Objekts. Benutzen Sie ihr Variogrammmodell aus der vorangegangenen Übung und notieren Sie ihre R-Syntax im Protokoll.

```

#library(gstat)
#?krige()
#as(object, 'SpatialGridDataFrame')
#krig <-krige(SPDF@data$Ca_exch~1,
#             model= v_fit,
#             #locations = SPDF,
#             #data= PixDF,
#             )

# krig <- gstat::krige(formula = Ca_exch~1, #~x+y
#                       locations = SPDF,
#                       data= PixDF,
#                       model= v_fit
#                       )

```

13 c)

Welche zwei Daten-Attribute produziert die krig-Funktion und wofür stehen sie?

```
#str(krig)
```

pred: Vorhersage Wert in Einheit der Ausgangsgröße var: Varianz der Vorhersage

13 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.

```
var(SPDF@data$Ca_exch) #Varianz der Ca_exch Werte = 123.8529
```

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

13 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.

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.

```
##?Palettes
library(lattice)
library(RColorBrewer)

# spplot(krig,
#         main= "Interpolation austauschbarer Ca-Ionen"
#         #,
#         )
```

13 f)

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

Literatur

Hengl, Tomislav. 2006. "Finding the Right Pixel Size." *Computers & Geosciences* 32 (9): 1283–98. <https://doi.org/10.1016/j.cageo.2005.11.008>.