Aufgabenzettel 05

Gruppe 01

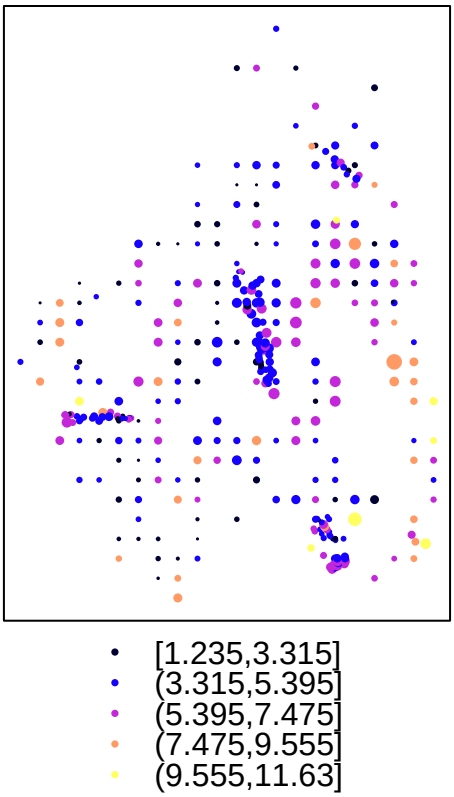
02.06.2020

Aufgabe 12 Die große Variogrammmodellierung

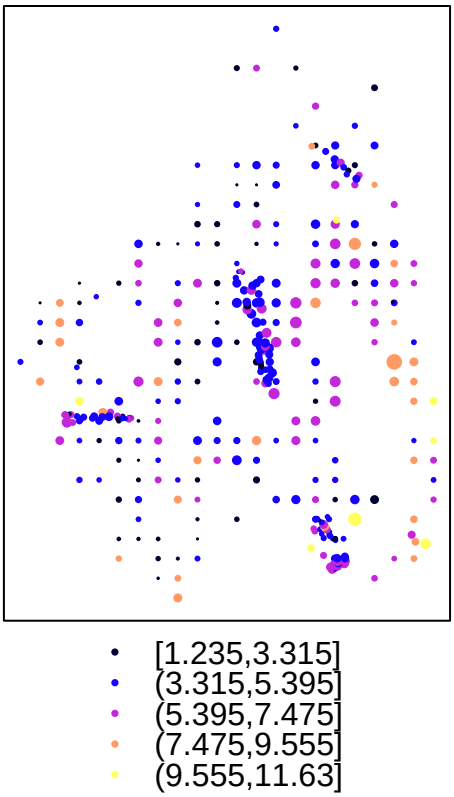
12 a)

Laden Sie das Paket gstat und plotten Sie die Variogrammwolke für die austauschbaren Ca-Ionen. Beschreiben Sie in ein, zwei Sätzen die Grundstruktur des Plots und erklären Sie ebenso präzise das Zustandekommen vereinzelter vertikaler Streifen, z.B. bei 300, 600 oder 750m.

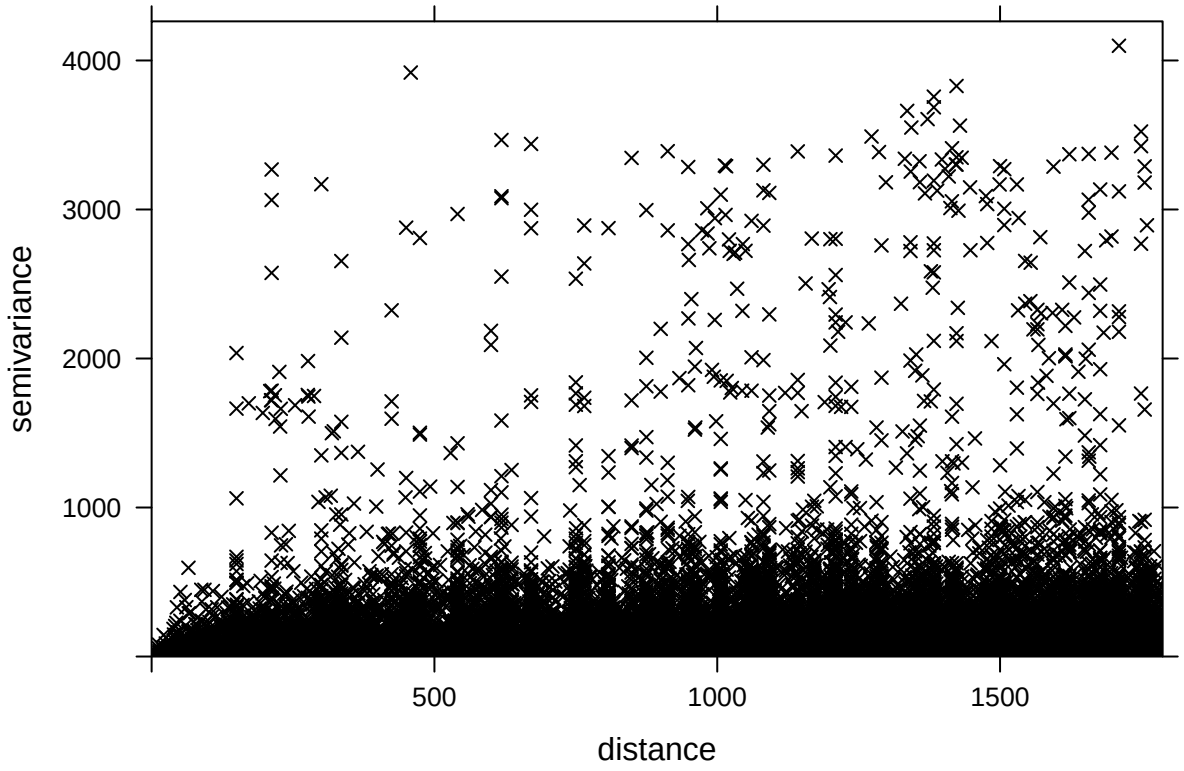
```
#install.packages("gstat")
#install.packages("sp")
library(sp)
library(gstat)

# Methode 1
SPDF1 <- ljz
coordinate <- SpatialPoints(SPDF1[c("EAST", "NORTH")])
coordRefSys <- CRS("+init=epsg:32650")
SPDF1 <- SpatialPointsDataFrame(coords = coordinate,
                                data = ljz,
                                proj4string = coordRefSys # funktioniert nicht? Warum?
                                 )
proj4string(SPDF1) <- CRS("+init=epsg:32650")

spplot(obj = SPDF1, zcol= 7, cex = sqrt(SPDF1$Ca_exch)/10)
```




```
#Methode 2
SPDF2 <- ljz
coordinates(SPDF2) <- ~ EAST+NORTH
proj4string(SPDF2) <- CRS("+init=epsg:32650")
spplot(obj = SPDF2, zcol= 5, cex = sqrt(SPDF2$Ca_exch)/10)
```




```
#?variogram()
cloud <- variogram(Ca_exch~1,
                   SPDF2,
                   #cutoff= 2202,
                   #width= 1000,
                   cloud = TRUE)
plot(cloud, pch=4, col="black")
```



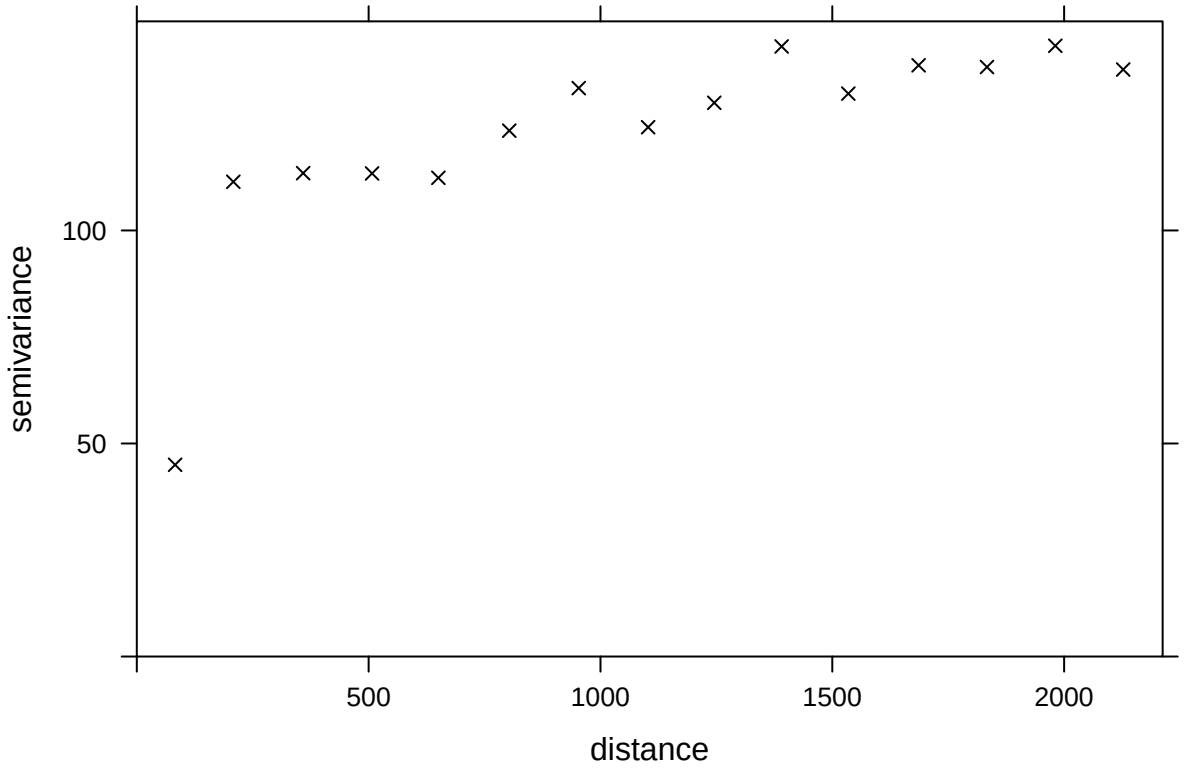

```
length(cloud$np)
```

```
## [1] 35827
```

12 b)

Erstellen Sie nun ein omnidirektionales empirisches Variogramm für die austauschbaren Ca-Ionen. Ändern Sie dabei die Default-Einstellungen der Argumente cutoff und width, um ein aussagekräftiges Ergebnis zu erzielen. Beziehen Sie sich auf gültige 'Daumenregeln' und begründen Sie knapp ihre Parameterwahl. Plotten Sie ihr Variogramm; verwenden Sie Kreuze anstelle von Punkten.

```
var1 <- variogram(Ca_exch~1,
                  SPDF2,
                  cutoff=2202,
                  width= (2202/15)
                    )

plot(var1, pch=4, col="black")
```




```
str(var1)
```

```
## Classes 'gstatVariogram' and 'data.frame':   15 obs. of  6 variables:
##  $ np     : num  941 1605 2023 2120 2478 ...
##  $ dist   : num  82.6 208.2 358.8 507.4 650.4 ...
##  $ gamma  : num  45 111 113 113 112 ...
##  $ dir.hor: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ dir.ver: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ id     : Factor w/ 1 level "var1": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "direct")='data.frame':   1 obs. of  2 variables:
##   ..$ id       : Factor w/ 1 level "var1": 1
##   ..$ is.direct: logi TRUE
##  - attr(*, "boundaries")= num  0 147 294 440 587 ...
##  - attr(*, "pseudo")= num 0
##  - attr(*, "what")= chr "semivariance"
```

12 c)

Schauen Sie sich das Objekt, welches durch die Methode variogram erzeugt wird, etwas genauer an. Welchen Klassen gehört es an? Wofür stehen die ersten drei Variablen des erzeugten Objekts?

```
str(var1)
```

```
## Classes 'gstatVariogram' and 'data.frame':   15 obs. of  6 variables:
##  $ np     : num  941 1605 2023 2120 2478 ...
##  $ dist   : num  82.6 208.2 358.8 507.4 650.4 ...
##  $ gamma  : num  45 111 113 113 112 ...
##  $ dir.hor: num  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ dir.ver: num  0 0 0 0 0 0 0 0 0 0 0 ...
##  $ id     : Factor w/ 1 level "var1": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "direct")='data.frame':   1 obs. of  2 variables:
##   ..$ id       : Factor w/ 1 level "var1": 1
##   ..$ is.direct: logi TRUE
##  - attr(*, "boundaries")= num  0 147 294 440 587 ...
##  - attr(*, "pseudo")= num 0
##  - attr(*, "what")= chr "semivariance"
```

Aus ?variogram() np: the number of point pairs for this estimate; in case of a variogramCloud see below

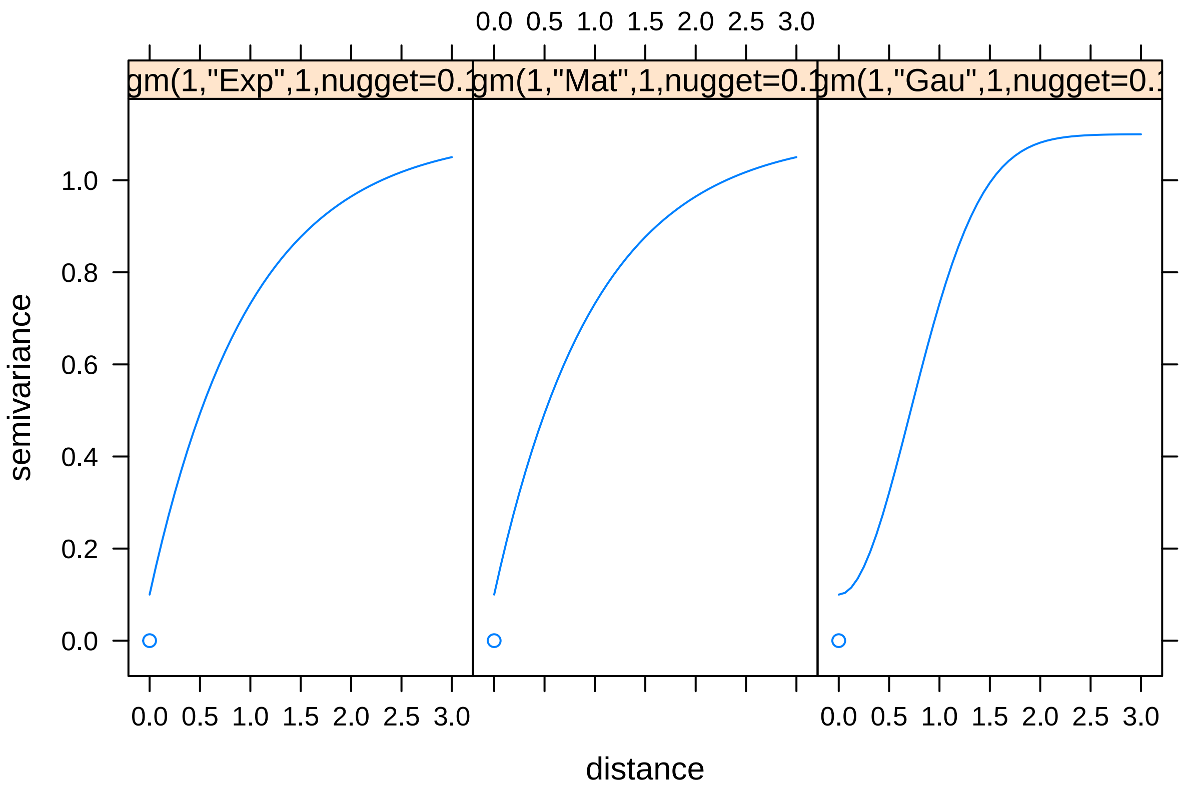
dist: the average distance of all point pairs considered for this estimate

gamma: the actual sample variogram estimate

12 d)

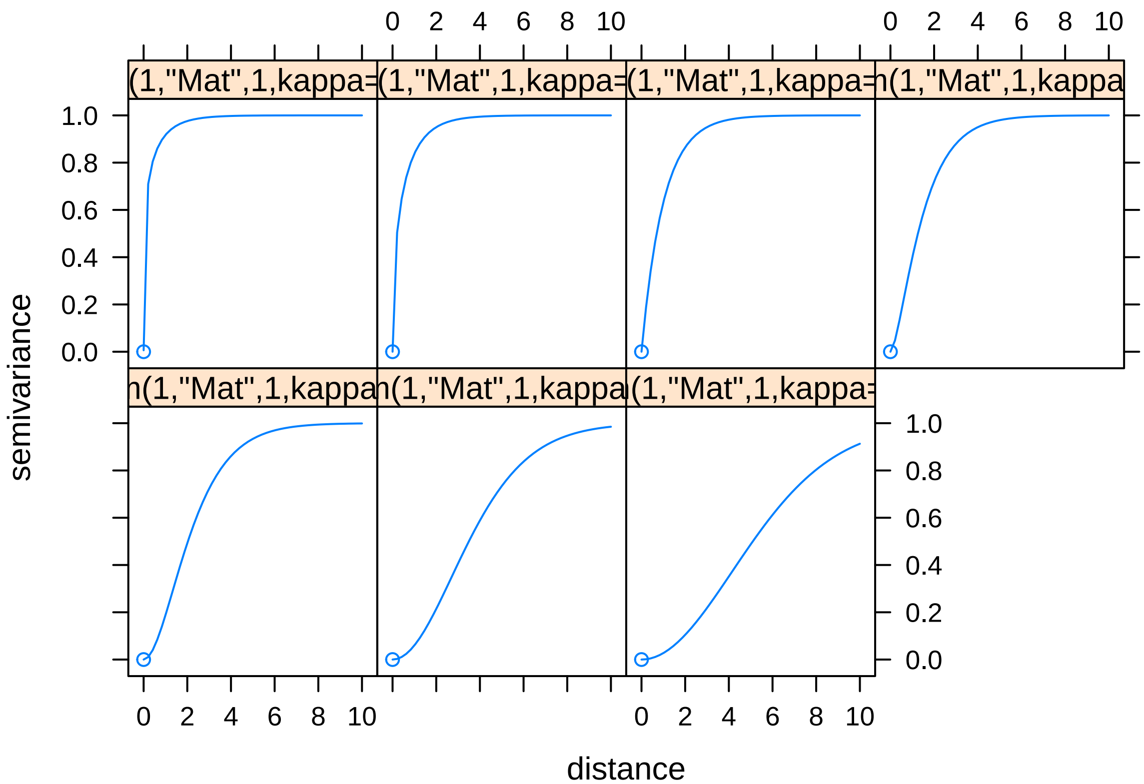
Fügen Sie ihrem empirischen Variogramm-Plot aus 12 b) ein passendes Modell hinzu. Erläutern Sie kurz, wie Sie vorgegangen sind. Hinweis: Mit dem Befehl show.vgms listen Sie die in gstat verfügbaren, autorisierten Modelle auf.

```r
# ?vgm()
# ?fit.variogram()

m <- vgm(psill = 150,
         model = c("Exp", "Mat", "Gau"),
         range = 2202,
         nugget = 50)

# v_fit <- fit.variogram(object = var1,
#                        model = m,
#                        fit.sills = ,
#                        fit.ranges = ,
#                        fit.method = ,
#                          )
#plot(v_fit)

# ?show.vgms
# show.vgms()



show.vgms(models = c("Exp", "Mat", "Gau"), nugget = 0.1)
```




```r
# show a set of Matern models with different smoothness:
show.vgms(kappa.range = c(.1, .2, .5, 1, 2, 5, 10), max = 10)
```

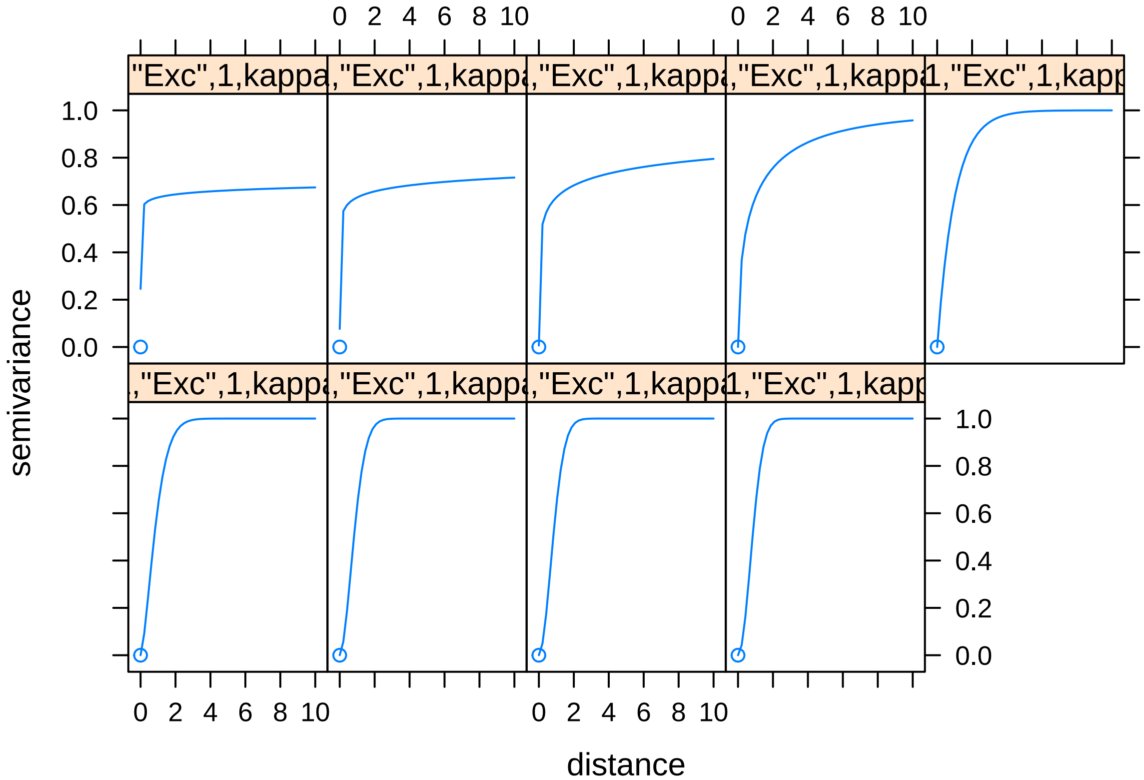
```
## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length
```




```r
# show a set of Exponential class models with different shape parameter:
show.vgms(kappa.range = c(.05, .1, .2, .5, 1, 1.5, 1.8, 1.9, 2), models = "Exc", max = 10)
```

```
## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length
```

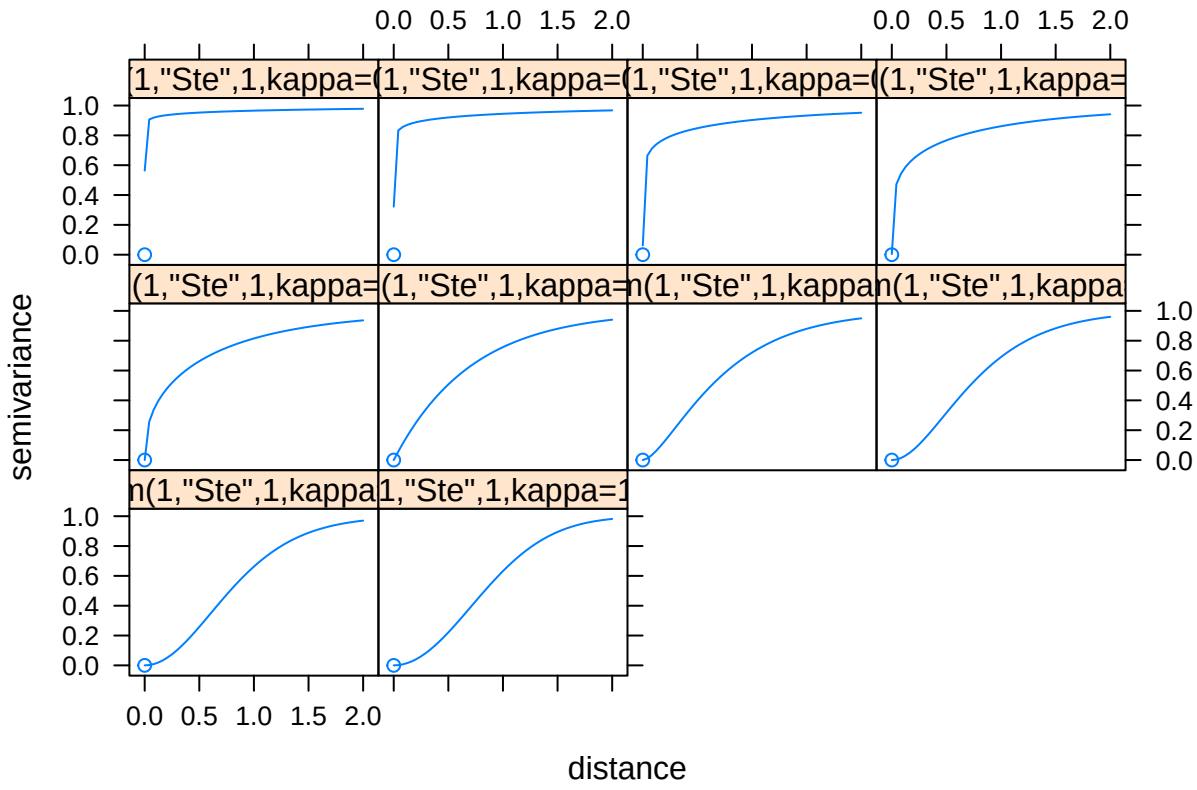
```
## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length
```




```r
# show a set of models with different shape parameter of M. Stein's representation of the Matern:
show.vgms(kappa.range = c(.01, .02, .05, .1, .2, .5, 1, 2, 5, 1000), models = "Ste", max = 2)
```

```
## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length
```

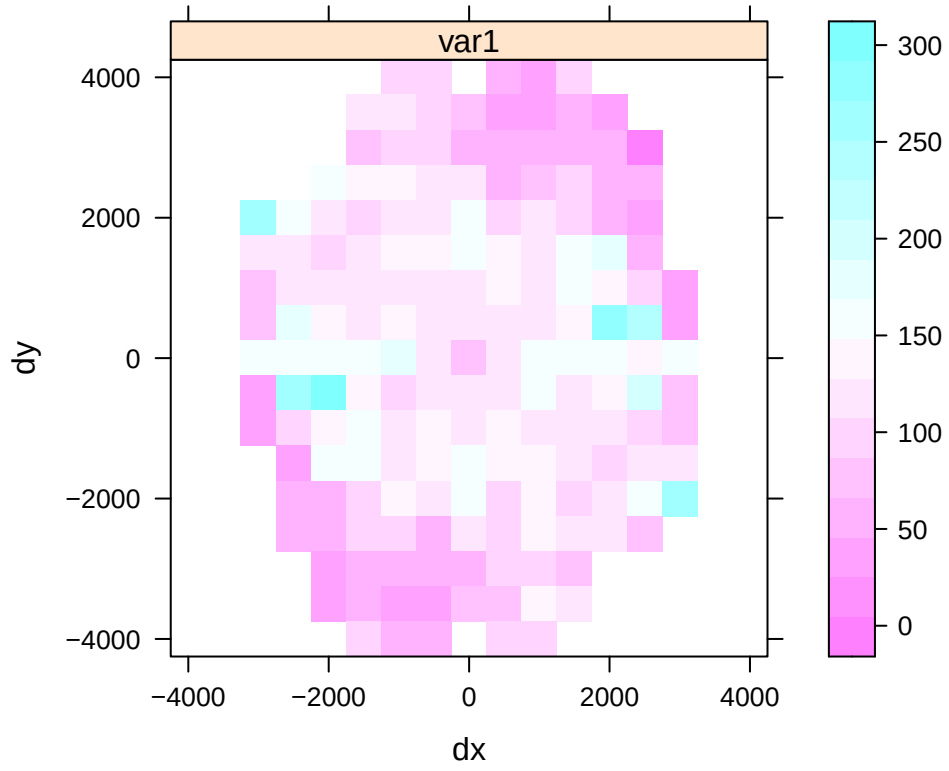
```
## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length

## Warning in v.level[(i * n + 1):((i + 1) * n)] <- rep(m.name, n): number of items
## to replace is not a multiple of replacement length
```



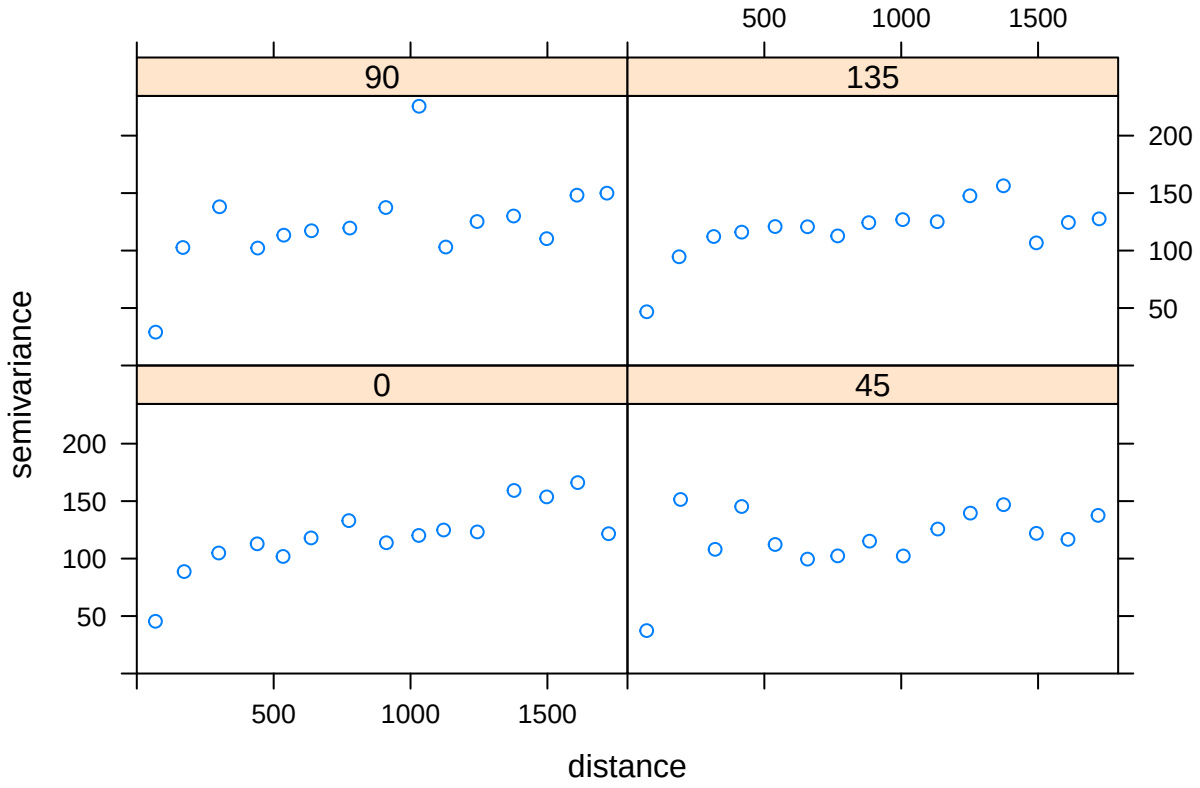
12 e)

Betrachten Sie noch einmal ihre Variogramme aus b) bzw. d) und interpretieren Sie diese. Argumentieren Sie auf Basis der Variogrammcharakteristik und des Nugget-To-Sill-Ratio nach Cambardella et al. (1994) (vgl. S.1508f.). Wie bewerten Sie das autokorrelative Verhalten der Zielvariablen im Raum?

```
var_map <- variogram(Ca_exch~1,
                     SPDF2,
                     cutoff= 4000,
                     width= 500,
                     map= TRUE
                     )

var_aniso <- variogram(Ca_exch~1, SPDF2, alpha=c(0,45,90,135))
plot(var_map)
```




```
plot(var_aniso)
```



Literatur

Cambardella, C. A., T. B. Moorman, J. M. Novak, T. B. Parkin, D. L. Karlen, R. F. Turco, and A. E. Konopka. 1994. "Field-Scale Variability of Soil Properties in Central Iowa Soils." *Soil Science Society of America Journal* 58 (5): 1501–11. https://doi.org/10.2136/sssaj1994.03615995005800050033x.