

# Aufgabenzettel 01

Gruppe 01

27.04.2020

## Aufgabe 1

### 1 a)

Recherchieren sie Pakete, die fuer geostatistische Aufgabenstellungen relevant sein koennten. Nennen sie dabei mindestens zwei und beschreiben sie kurz deren wesentliche Ziele. Nutzen sie die R-Homepage <http://cran.r-project.org/> (<http://cran.r-project.org/>) fuer ihre Suche aus.

- gstat: Das gstat-Paket beinhaltet Funktionen mit denen verschiedene Modellierungen, das Erstellen von Variogrammen und Kriging moeglich sind. <https://cran.r-project.org/web/packages/gstat/gstat.pdf> (<https://cran.r-project.org/web/packages/gstat/gstat.pdf>)
- spANOVA: ermoeeglicht eine Varianzanalyse von korrelierenden Rasterdatensaetzen <https://cran.r-project.org/web/packages/spANOVA/spANOVA.pdf> (<https://cran.r-project.org/web/packages/spANOVA/spANOVA.pdf>)
- ggmap: Erweitert das Plotting-Paket ggplot2 um Kartendarstellungen <https://cran.r-project.org/web/packages/ggmap/ggmap.pdf> (<https://cran.r-project.org/web/packages/ggmap/ggmap.pdf>)
- rgdal: R's Schnittstelle zur C/C++ Bibliothek gdal <https://cran.r-project.org/web/packages/rgdal/rgdal.pdf> (<https://cran.r-project.org/web/packages/rgdal/rgdal.pdf>)
- rgeos: R's Schnittstelle Vektor Bibliothek geos <https://cran.r-project.org/web/packages/rgeos/rgeos.pdf> (<https://cran.r-project.org/web/packages/rgeos/rgeos.pdf>)
- maptools: Bietet diverse Funktionen zum manipulieren von Geodaten und Erstellung von Kartengraphiken <https://cran.r-project.org/web/packages/maptools/maptools.pdf> (<https://cran.r-project.org/web/packages/maptools/maptools.pdf>)
- tmap: Paket zum schnellen erstellen von thematischen Karten. <https://cran.r-project.org/web/packages/tmap/tmap.pdf> (<https://cran.r-project.org/web/packages/tmap/tmap.pdf>)

### 1 b)

Erforschen sie selbstaendig die Funktion `read_delim` des `readr` Paketes. Beschreiben sie kurz, was die Methode macht und auf welche Einstellungsmoeglichkeiten (Argumente) besonders zu achten ist.

Die `read_delim`-Funktion im `readr`-Packet ermoeeglicht das Einlesen von Tabellen aus Textdokumenten als tibble (tidyverse data.frame). Dazu muessen die einzelnen Spalten als solche erkannt werden, was in der Regel durch das erkennen von Trennzeichen wie Tab, Komma oder Semikolon erfolgt.

Im wesentlichen wird zuerst fuer das Argument `file` der Dateipfad zur gewuenschten Datei angegeben. Fuer das Argument `delim` beschreibt, wodurch oder mit welchem Zeichen einzelne Felder mit Inhalt getrennt werden und `quote` mit welchem Zeichen strings, also Text, markiert wird. Darueber hinaus wird mit dem Argument `col_names=True/False` festgelegt, ob die erste Zeile der Datei Spaltennamen darstellen oder nicht. Mit weiteren Argumenten, kann festgelegt werden, ob Zeilen beim einlesen uebersprungen werden sollen, Min- und Maxwerte beschrieben oder ob leere Zeilen uebersprungen werden sollen.

<https://cran.r-project.org/web/packages/readr/readr.pdf> (<https://cran.r-project.org/web/packages/readr/readr.pdf>) S. 25, 26

# Aufgabe 2 Vektor und Matrizen

## 2 a) working directory

Oeffnen Sie die .rmd mit RStudio und ueberpruefen Sie in der Konsole, ob Sie sich im neu eingerichteten Verzeichnis befinden. Ist dies nicht der Fall wechseln sie in das Verzeichnis.

```
getwd()
```

```
## [1] "C:/docs/wd"
```

```
list=ls()  
print(list)
```

```
## character(0)
```

```
rm(list=ls())  
setwd("C:/docs/wd")  
#load("C:/docs/wd/z01_Geostatistik_s20_Gr01.Rmd") oeffnet Datei, wenn diese zuvor mit save.im  
age() gespeichert worden ist  
file.edit("C:/docs/wd/z01_Geostatistik_s20_Gr01.Rmd")
```

## 2 b)

Vektor v1 der Laenge 15 mit einer Schrittweite von 0.5, beginnend bei 0

```
v1 = seq(from=0, by=0.5 , length.out=15)  
print(v1);
```

```
## [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0
```

## 2 c)

Bestimmen sie die Funktionswerte fuer  $f(v1) = 3 * v1 + ihre\_gruppen\_id$  und fassen sie v1 sowie f(v1) in einer Matrix (15 x 2) M1 zusammen (cbind)

*#Funktion f, die  $f(v1) = 3 * v1 + 01$  erfuehlt*

```
f <- function(a) {
  f <- 3.0*a+1.0
  return(f)
}
```

```
v2 = f(v1)
```

```
M1 = cbind(v1, v2)
```

```
print(M1);
```

```
##          v1    v2
## [1,] 0.0  1.0
## [2,] 0.5  2.5
## [3,] 1.0  4.0
## [4,] 1.5  5.5
## [5,] 2.0  7.0
## [6,] 2.5  8.5
## [7,] 3.0 10.0
## [8,] 3.5 11.5
## [9,] 4.0 13.0
## [10,] 4.5 14.5
## [11,] 5.0 16.0
## [12,] 5.5 17.5
## [13,] 6.0 19.0
## [14,] 6.5 20.5
## [15,] 7.0 22.0
```

## 2 d)

Erzeugen sie die transponierte Matrix M1t und multiplizieren sie diese mit M1 zu M3. Lassen sie sich das resultierende Objekt M3 anzeigen. Loeschen sie anschließend das Objekt v1 und speichern sie den aktuellen Workspace sowie die R-History in ihrem Kursordner ab.

```
M1t = t(M1)
```

```
print(M1t)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## v1    0  0.5   1  1.5   2  2.5   3  3.5   4  4.5   5  5.5   6  6.5
## v2    1  2.5   4  5.5   7  8.5  10 11.5  13 14.5  16 17.5  19 20.5
##      [,15]
## v1        7
## v2       22
```

```
M3 = M1%*%M1t
```

```
print(M3);
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,]  1.0  2.5  4.0  5.5  7.0  8.5 10.0 11.5 13.0 14.5 16.0 17.5
## [2,]  2.5  6.5 10.5 14.5 18.5 22.5 26.5 30.5 34.5 38.5 42.5 46.5
## [3,]  4.0 10.5 17.0 23.5 30.0 36.5 43.0 49.5 56.0 62.5 69.0 75.5
## [4,]  5.5 14.5 23.5 32.5 41.5 50.5 59.5 68.5 77.5 86.5 95.5 104.5
## [5,]  7.0 18.5 30.0 41.5 53.0 64.5 76.0 87.5 99.0 110.5 122.0 133.5
## [6,]  8.5 22.5 36.5 50.5 64.5 78.5 92.5 106.5 120.5 134.5 148.5 162.5
## [7,] 10.0 26.5 43.0 59.5 76.0 92.5 109.0 125.5 142.0 158.5 175.0 191.5
## [8,] 11.5 30.5 49.5 68.5 87.5 106.5 125.5 144.5 163.5 182.5 201.5 220.5
## [9,] 13.0 34.5 56.0 77.5 99.0 120.5 142.0 163.5 185.0 206.5 228.0 249.5
## [10,] 14.5 38.5 62.5 86.5 110.5 134.5 158.5 182.5 206.5 230.5 254.5 278.5
## [11,] 16.0 42.5 69.0 95.5 122.0 148.5 175.0 201.5 228.0 254.5 281.0 307.5
## [12,] 17.5 46.5 75.5 104.5 133.5 162.5 191.5 220.5 249.5 278.5 307.5 336.5
## [13,] 19.0 50.5 82.0 113.5 145.0 176.5 208.0 239.5 271.0 302.5 334.0 365.5
## [14,] 20.5 54.5 88.5 122.5 156.5 190.5 224.5 258.5 292.5 326.5 360.5 394.5
## [15,] 22.0 58.5 95.0 131.5 168.0 204.5 241.0 277.5 314.0 350.5 387.0 423.5
##      [,13] [,14] [,15]
## [1,]  19.0  20.5  22.0
## [2,]  50.5  54.5  58.5
## [3,]  82.0  88.5  95.0
## [4,] 113.5 122.5 131.5
## [5,] 145.0 156.5 168.0
## [6,] 176.5 190.5 204.5
## [7,] 208.0 224.5 241.0
## [8,] 239.5 258.5 277.5
## [9,] 271.0 292.5 314.0
## [10,] 302.5 326.5 350.5
## [11,] 334.0 360.5 387.0
## [12,] 365.5 394.5 423.5
## [13,] 397.0 428.5 460.0
## [14,] 428.5 462.5 496.5
## [15,] 460.0 496.5 533.0
```

*#Löschen sie anschließend das Objekt v1 und speichern sie den aktuellen  
#Workspace sowie die R-History in ihrem Kursordner ab*

```
rm(v1);
```

```
#save(list = ls(all.names = TRUE), file = ".RData", envir = .GlobalEnv) gleichbedeutend wie s  
ave.image()  
#save.image()  
#save(mein_datensatz, file="zwischenstand_mein_datensatz.Rdata")  
#savehistory(".Rhistory")  
#quit() schließt das R-Skript
```

*#Der Befehl savehistory(".Rhistory") lässt sich im R-Code ausführen, führt aber zu Fehlermeldungen beim Knit. Dieser wird daher vor dem Knitvorgang auskommentiert. Das gleiche Problem liegt bei quit() vor.*

## Aufgabe 3

- Zufallsvariable (random variable): "Eine Abbildung oder Funktion, die den Elementen der Ergebnis- oder Ereignismenge eines Zufallsexperimentes reelle Zahlen zuordnet, heißt Zufallsvariable" (Hedderich and Sachs 2018). Sie ist nicht zufällig oder variabel, sondern durch reelle Zahlen festgelegt. Sie kann

diskret oder stetig sein. Wenn die Zufallsvariable diskret ist, kann sie nur bestimmte Werte annehmen, dessen Einzelwahrscheinlichkeiten 1 ergibt. Andernfalls ist sie stetig.

- Zufallsfunktion (random function): Definiert alle Punkte einer Region mithilfe von einfachen, charakterisierbaren Eigenschaften (Wackernagel 2003) Eine Zufallsfunktion  $Z$  wird durch die Verteilung der einzelnen Zufallsvariablen  $Z(x)$  an jeder Stelle  $x$  und die gegenseitigen Abhängigkeiten charakterisiert (Dutter 1985)
- Regionalisierte Variable (regionalized variable): Realisation einer Zufallsfunktion. Die Variable  $z(x)$  ist von einem Ort  $x$  in einer bestimmten Region abhängig (Dutter 1985). Eine Variable  $z(x)$ , die Werte in Abhängigkeit vom Ort  $x$  in einem bestimmten Bereich (Region) angibt, bezeichnet man als regionalisierte Variable (Dutter 1985)
- Stationarität (stationarity): Wenn Erwartungswert, Mittelwert und Varianz invariant gegen Veränderungen der Datengrundlage sind, ist eine Zufallsvariable stationär. Die Variable wäre dann "unabhängig von der absoluten Lage und konstant im Beobachtungsraum" (Akin and Siemes 1988).
- Semivarianz (semivariance): Als Semivarianz bezeichnet man in der Statistik die halbe, mittlere, quadrierte euklidische Distanz zwischen den Messwerten  $z(x_i)$  und  $z(x_i+h)$  an den Orten  $x_i$  und  $x_i+h$  für den Abstand bzw. Vektor  $h$ . (Shine and Wakefield 1999) Beschreibt zwei Pixel in Relation zu einander in Bezug auf ihre Distanz unter der Annahme, dass je größer die Entfernung ist, desto geringer ist ihre Ähnlichkeit. Dabei wird von einem Pixel ausgehend der Zusammenhang zu den umliegenden Pixeln beschrieben (Curran 1988).

## Literatur

Akin, H., and H. Siemes. 1988. *Praktische Geostatistik: Eine Einführung Für Den Bergbau Und Die Geowissenschaften*. Springer.

Curran, P. J. 1988. "The Semivariogram in Remote Sensing: An Introduction." *Remote Sensing of Environment* 24 (3): 493–507.

Dutter, R. 1985. "Regionalisierte Variable." *Geostatistik. Mathematische Methoden in Der Technik*. 2: 55–60.

Hedderich, J., and L. Sachs. 2018. *Angewandte Statistik*. Springer.

Shine, J. A., and G. I. Wakefield. 1999. *A Comparison of Supervised Imagery Classification Using Analyst-Chosen and Geostatistically-Chosen Training Sets*.

Wackernagel, H. 2003. "Regionalized Variable and Random Function." *Multivariate Geostatistics*, 39–44.