Aufgabe 08

Gruppe 01

23.06.2020

Aufgabe 17

17 a)

Für die heutige Sitzung steht Ihnen eine gezippte Datei zur Verfügung. Nutzen Sie R, um die Datei „data.zip“ zu entpacken. Im Anschluss können Sie mit den enthaltenen Daten weiterarbeiten.

```
unzip(zipfile = "data.zip", list = TRUE) # zeigt Inhalte der ZIP-file
```

```
##                    Name Length                Date
## 1             convg.csv 594323 2020-03-30 15:27:00
## 2               swi.csv 591747 2020-03-30 15:27:00
## 3 yingtan_20_ueb3.RData  11186 2020-03-30 15:27:00
## 4 yingtan_elevation.asc  64336 2020-03-30 15:27:00
```

```
unzip(zipfile = "data.zip")
```

17 b)

Für das von Ihnen untersuchte Gebiet liegt Ihnen ein Höhenmodell als ASCII Datei vor. Für weitere Analysen wollen Sie dieses nun in R einlesen. Machen Sie sich also mit dem „raster“ Paket vertraut und lesen Sie mit der entsprechenden Funktion die Datei „yingtan_elevation.asc“ in Ihre Arbeitsumgebung ein und vergeben Sie das passende geographische Referenzsystem.

```
elevation <- read.asciigrid(fname = "yingtan_elevation.asc" ,proj4string = CRS("+init=epsg:32650"))
elevation <- raster(elevation)
```

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

```
#str(elevation)
#plot(elevation, main= "Elevation")
```

Aufgabe 18

18 a)

a) Ihre nächste Aufgabe ist es, aus dem Höhenmodell die bekannten Reliefparameter Hangneigung, Hangneigungsrichtung, Terrain Ruggedness Index, Topographic Position Index, Roughness und die Flow Direction abzuleiten. Nutzen Sie dazu die entsprechende Funktion des „raster“ Paketes. Begründen Sie die Wahl der Anzahl berücksichtigter Nachbarn.

```
#Hangneigung
#Hangneigungsrichtung
#Terrain Ruggedness Index
#Topographic Position Index
```

```
#Roughness
#Flow Direction
raster.1 <- terrain(elevation,
                    opt=c("slope", "aspect", "TPI", "TRI", "roughness", "flowdir"),
                    unit="degrees",
                    neighbors=4, #queens vs Rooks case
                    #filename=
                    )
crs(raster.1) <- CRS("+init=epsg:32650")
#str(raster.1)
```

Begründung zu Wahl der Nachbarn: ...

18 b)

Unvermittelt teilt Ihr Kollege Ihnen mit, dass er soeben den Konvergenz-/Divergenz-Index (convg.csv) sowie den Saga Wetness-Index (swi.csv) in SAGA berechnet hat. Er gibt Ihnen die Raster als XYZ Datei im .csv Format. Lesen Sie diese ebenfalls in Ihre Arbeitsumgebung ein, wandeln Sie diese in „raster“ um und verbinden Sie alle Reliefparameter inklusive des Höhenmodells in einem RasterStack. Beschreiben Sie den Unterschied zwischen einem RasterStack und einem RasterBrick in knappen Worten

```
#Konvergenz-/Divergenz-Index
convg <- read.csv2("convg.csv", header = TRUE, sep = ";")

# convg <- rasterize(
#                     x=convg$x,
#                     y=convg$y,
#                     field=convg$CONVG2,
#                     background=NA
#                      )

convg <- rasterFromXYZ(convg,
                       #res=c(NA,NA),
                       crs=CRS("+init=epsg:32650"),
                       #digits=10
                       )
crs(convg) <- CRS("+init=epsg:32650")

#Saga Wetness-Index
SWI <- read.csv2("swi.csv", header = TRUE, sep = ";")
SWI <- rasterFromXYZ(SWI,
                       #res=c(NA,NA),
                       crs=CRS("+init=epsg:32650"),
                       #digits=10
                       )
crs(SWI) <- CRS("+init=epsg:32650")

#Rasterstack
yingtan.stack <- stack(elevation, raster.1, convg, SWI)
# test <- as.list(yingtan.stack)
# test
```

Unterschied RasterStack und RasterBrick: Die Objekttypen R RasterStack und RasterBrick können beide mehrere Bänder speichern. Wie sie jedes Band speichern, ist jedoch unterschiedlich. Die Bänder in einem RasterStack werden als “Links” zu Rasterdaten gespeichert, die sich auf dem PC befinden. Ein RasterBrick

enthält alle Objekte, die innerhalb des eigentlichen R-Objekts gespeichert sind. In den meisten Fällen kann man mit einem RasterBrick auf die gleiche Weise arbeiten wie mit einem RasterStack. Allerdings ist eine RasterBrick oft effizienter und schneller zu verarbeiten - was bei der Arbeit mit größeren Dateien wichtig ist.

18 c)

Ihnen fällt auf, dass Sie den Stack in weiser Voraussicht auf den kommenden Übungszettel zusätzlich in das Ihnen länger bekannten SpatialGridDataFrame Format überführen wollen. Erzeugen Sie entsprechend eine Variable diesen Dateityps aus dem soeben erzeugten RasterStack.

```
#SpatialGridDataFrame
SGDF <- yingtan.stack
SGDF <- as(SGDF,"SpatialGridDataFrame") #"SpatialPixelsDataFrame"

#str(SGDF)
```

18 d)

Machen Sie sich nun mit den Reliefparametern vertraut. Beschreiben Sie jeden Parameter kurz.

Hangneigung/slope : Hangneigungsrichtung/ slope direction : Terrain Ruggedness Index : Topographic Position Index : Roughness : Flow Direction :

Aufgabe 19

19 a)

Für die folgenden Arbeiten ist es für Sie von Interesse, welche Wertausprägungen die Reliefparameter an den Beprobungstandorten annehmen. Fügen Sie entsprechend die Werte der jeweiligen Zellen des RasterStacks der Tabelle des SpatialPointsDataFrames hinzu. Suchen Sie in der Hilfe des „raster“ Paketes nach einer entsprechenden Funktion.

```
# Erzeugen von SpatialPointsDataFrame
load("C:/docs/github/CAU_Geostatistik/Aufgabe_8/yingtan_20_ueb3.RData")
SPDF <- ljz
coordinates(SPDF) <- ~ EAST+NORTH
proj4string(SPDF) <- CRS("+init=epsg:32650")
str(SPDF)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
##   ..@ data       : tibble [335 x 7] (S3: tbl_df/tbl/data.frame)
##   .. ..$ OBJECTID: int [1:335] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ SAMPLING: Factor w/ 10 levels "catA","catF",..: 10 10 10 10 10 10 10 10 10 10 ...
##   .. ..$ C       : num [1:335] 0.599 0.647 0.527 0.812 0.756 ...
##   .. ..$ Ca_exch : num [1:335] 12.38 13.13 3.77 31.69 24 ...
##   .. ..$ Mg_exch : num [1:335] 3.2 4.77 1.45 7.72 8.33 ...
##   .. ..$ K_exch  : num [1:335] 3.509 0.992 4.046 2.655 4.02 ...
##   .. ..$ Na_exch : chr [1:335] "0.452" "0.592" "0.122" "0.687" ...
##   ..@ coords.nrs : int [1:2] 3 4
##   ..@ coords     : num [1:335, 1:2] 490591 490591 490591 490741 490741 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:335] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:2] "EAST" "NORTH"
##   ..@ bbox       : num [1:2, 1:2] 490441 3121290 493591 3125630
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "EAST" "NORTH"
```

```
##   .. .. ..$ : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. ..@ projargs: chr "+proj=utm +zone=50 +datum=WGS84 +units=m +no_defs"
```

```
# Werte hinzufügen
ex <- extract(x = yingtan.stack, #nicht das richtige
              y = SPDF,
              )
```

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

19 b)

Erzeugen Sie zu guter Letzt einen anschaulichen Plot, der die Lage der Punkte vor dem Höhenmodell zeigt.