

R Analysis/Shiny

Dr. Benedikt Gräler

3rd September 2018

Agenda

- ▶ brief introduction
- ▶ few basics in R
- ▶ connecting with the sensorweb
- ▶ analysis of sensor data
- ▶ shiny apps

About myself

- ▶ Diploma in Mathematics @ WWU
- ▶ PhD in Geoinformatics (Geostatistics) @ ifgi
- ▶ PostDoc at Hydrology Dep. @ RUB
- ▶ since May 2017 working at @ 52N:
 - ▶ R development
 - ▶ Spatio-Temporal Data Science
 - ▶ Geostatistics

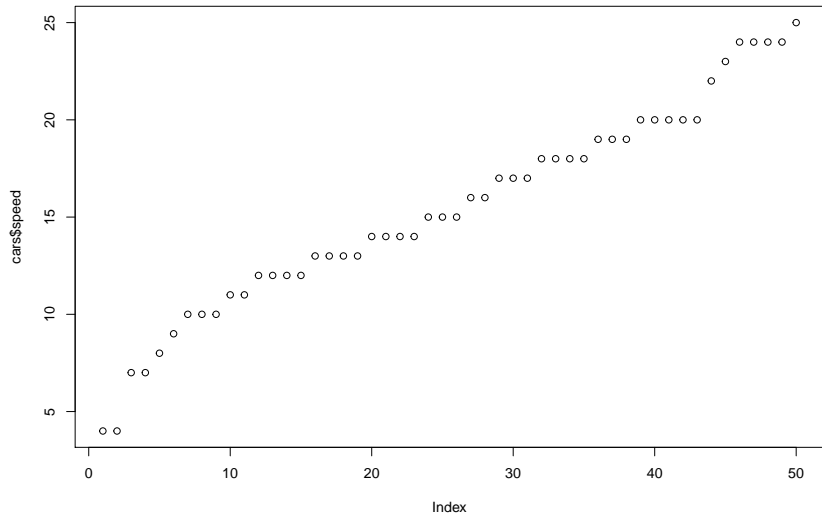
About R

- ▶ Scripting language with lots of build in statistics
- ▶ extensible via packages (CRAN: +10.000 pkgs, GitHub, ...)

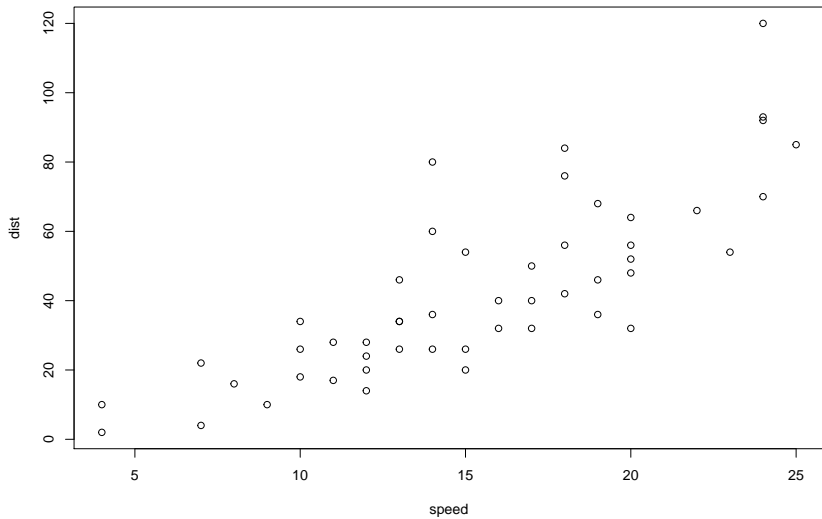
```
summary(cars)
```

```
##      speed      dist
## Min.   : 4.0    Min.    : 2.00
## 1st Qu.:12.0    1st Qu.: 26.00
## Median :15.0    Median : 36.00
## Mean   :15.4    Mean    : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
## Max.   :25.0    Max.    :120.00
```

```
plot(cars$speed)
```



```
plot(dist ~ speed, cars)
```



```
linMod <- lm(dist ~ speed, cars)
linMod
```

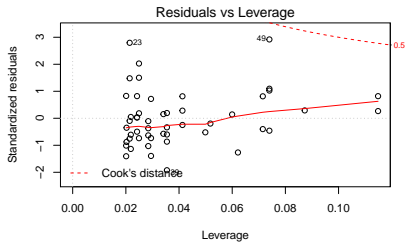
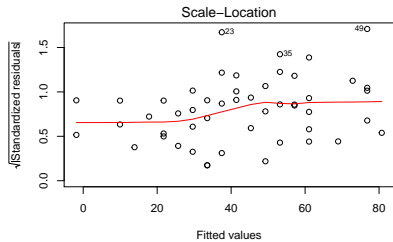
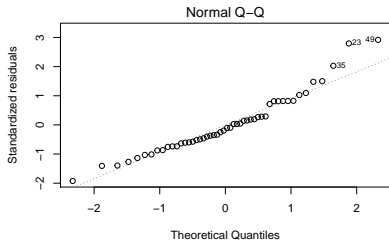
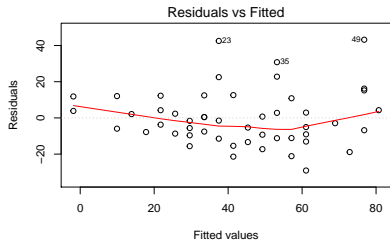
```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Coefficients:
## (Intercept)      speed
##    -17.579      3.932
```

```
summary(linMod)
```

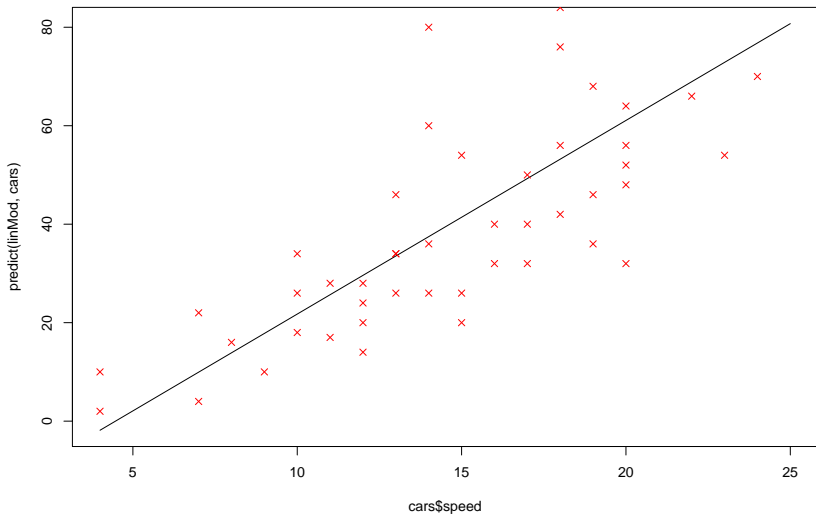
```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601  0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```



```
plot(linMod)
```



```
plot(predict(linMod, cars) ~ cars$speed, type="l")
points(dist ~ speed, cars, col="red", pch=4)
```



Connecting with the sensorweb

```
library(devtools)
install_github("52North/sos4R")
```

```
library(sos4R)
```

```
## Loading required package: XML
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
## Loading required package: sp
```

```
sos <- SOS(url = "http://sensorweb.demo.52north.org:80/PegelOnlineSOSv2.1/sos", switchCoordinates = TRUE)
```

```
## Warning in FUN(X[[i]], ...): Switching coordinates in envelope of
## ObservationOffering!
```

```
## Warning in FUN(X[[i]], ...): Switching coordinates in envelope of
## ObservationOffering!
```

```
## Warning in FUN(X[[i]], ...): Switching coordinates in envelope of
## ObservationOffering!
```

```
## Warning in FUN(X[[i]], ...): Switching coordinates in envelope of
## ObservationOffering!
```

```
## Warning in FUN(X[[i]], ...): Switching coordinates in envelope of
## ObservationOffering!
```

```
## Warning in FUN(X[[i]], ...): Switching coordinates in envelope of
```

```
str(sos@capabilities, 2)
```

```
## Formal class 'SosCapabilities_1.0.0' [package "sos4R"] with 8 slots
## ..@ filterCapabilities:Formal class 'SosFilter_Capabilities' [package "sos4R"] with 4 slots
## ..@ identification :Formal class 'OwsServiceIdentification' [package "sos4R"] with 8 slots
## ..@ provider :Formal class 'OwsServiceProvider' [package "sos4R"] with 3 slots
## ..@ operations :Formal class 'OwsOperationsMetadata' [package "sos4R"] with 4 slots
## ..@ contents :Formal class 'SosContents' [package "sos4R"] with 2 slots
## ..@ version : chr "1.0.0"
## ..@ updateSequence : chr NA
## ..@ owsVersion : chr "1.1.0"
```

```
foi <- SosFeatureOfInterest(sos@capabilities@operations@operations$GetFeatureOfInterest@parameters$feature
```

```
obs <- getObservation(sos,
                      sos@capabilities@contents@observationOfferings[[2]],
                      featureOfInterest = foi,
                      eventTime = list(SosEventTimeLatest()))
str(obs, 2)
```

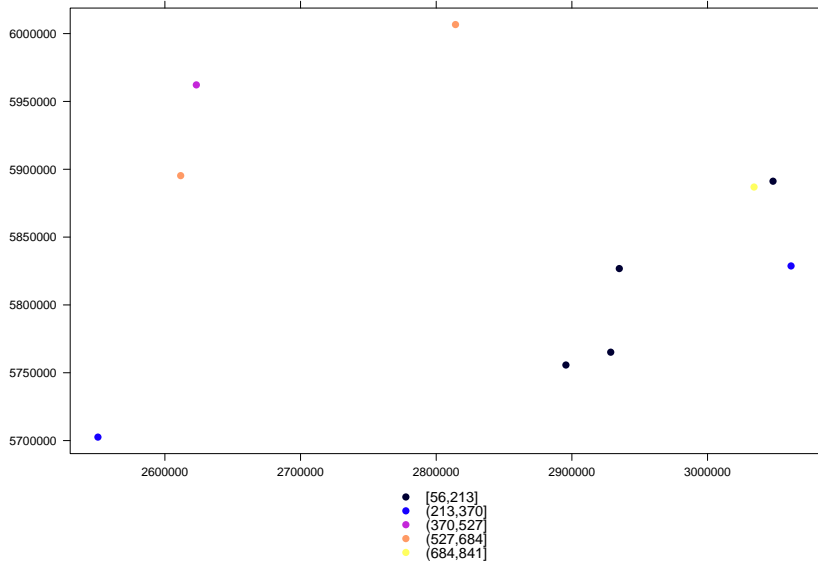
```
## [sos4R] Received response (size: 31688 bytes), parsing ...
## [sos4R] Finished getObservation to http://sensorweb.demo.52north.org:80/PegelOnlineSOSv2.1/sos
## --> received 10 observation(s) having 10 result values [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ].
```

```
## Formal class 'OmObservationCollection' [package "sos4R"] with 2 slots
## ..@ members :List of 10
## ..@ boundedBy:List of 3
```

```
obsSp <- as(obs, "SpatialPointsDataFrame")
str(obsSp, 2)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
## ..@ data      : 'data.frame': 10 obs. of 5 variables:
## ..@ coords.nrs : num(0)
## ..@ coords    : num [1:10, 1:2] 2895602 2550638 3034274 3061555 2928639 ...
## .. ..- attr(*, "dimnames")=List of 2
## ..@ bbox      : num [1:2, 1:2] 2550638 5702557 3061555 6006659
## .. ..- attr(*, "dimnames")=List of 2
## ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
```

```
spplot(obsSp, "Wasserstand", scales=list(arrows=FALSE))
```



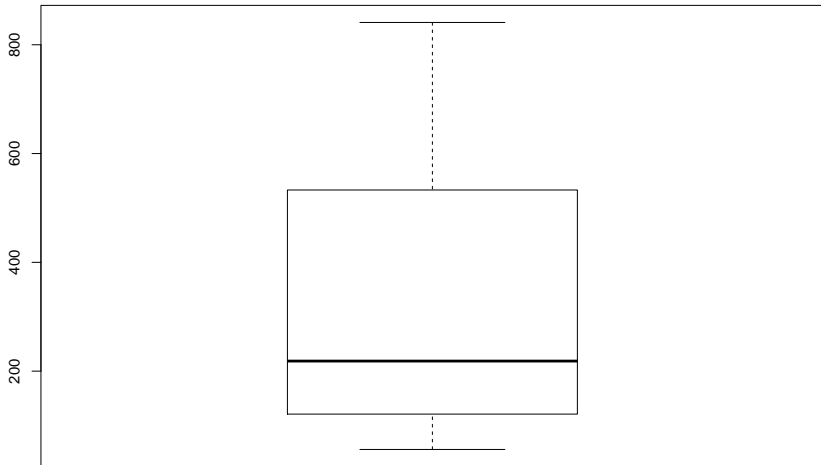
```
library(plotKML)  
plotKML(obsSp)
```

```
summary(obsSp@data$Wasserstand)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      56.0  141.2   218.5   329.8  504.0   841.0
```



```
boxplot(obsSp@data$Wasserstand)
```



```
library(fitdistrplus)
```

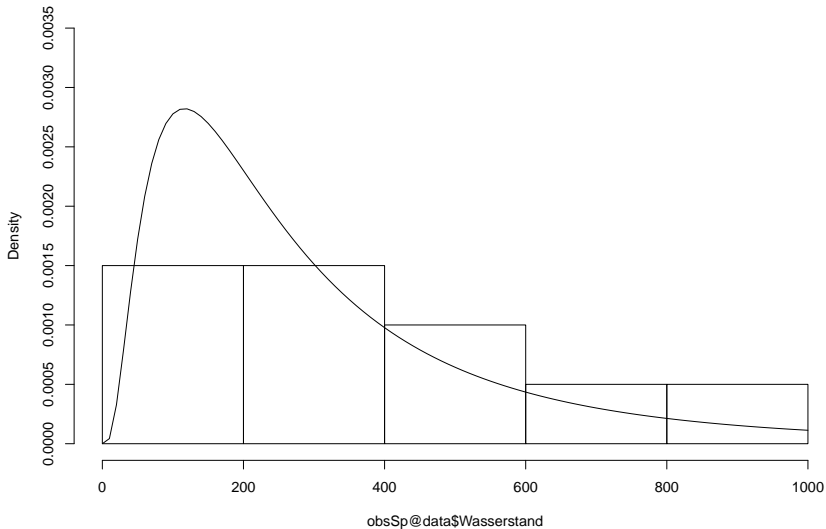
```
## Loading required package: MASS
```

```
## Loading required package: survival
```

```
fDist <- fitdist(obsSp@data$Wasserstand, "lnorm")  
dDist <- function(x) dlnorm(x, fDist$estimate[1], fDist$estimate[2])
```

```
hist(obsSp@data$Wasserstand, freq=FALSE, ylim=c(0,0.0035))  
curve(dDist, add=TRUE)
```

Histogram of obsSp@data\$Wasserstand



sensorweb4R

```
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
  
## The following object is masked from 'package:base':  
##  
##   date
```

```
library(xts)
```

```
## Loading required package: zoo  
  
##  
## Attaching package: 'zoo'  
  
## The following objects are masked from 'package:base':  
##  
##   as.Date, as.Date.numeric
```

```
library(spacetime)  
library(sensorweb4R)
```

```
## Warning: multiple methods tables found for 'geometry<-'  
  
## Welcome to sensorweb4R! The logging level is INFO - set it with 'flog.threshold(<level>, name = "sensorweb4R")'  
  
##  
## Attaching package: 'sensorweb4R'
```

```
sta.precip <- stations(endpoint, phenomenon = phe.all[5])
str(sta.precip, 2)
```

```
## Formal class 'Station' [package "sensorweb4R"] with 4 slots
## ..@ geometry:Formal class 'SpatialPoints' [package "sp"] with 3 slots
## ..@ id      : chr [1:28] "9" "42" "101" "4" ...
## ..@ label   : chr [1:28] "Barmen Wupperverband Hauptverwaltung" "Bever-Talsperre" "Beyenburg" "Burscl
## ..@ endpoint:Formal class 'Endpoint' [package "sensorweb4R"] with 2 slots
```

```
ts.precip <- timeseries(endpoint, station = sta.precip, phenomenon = phe.all[5])
str(ts.precip, 2)
```

```
## Formal class 'Timeseries' [package "sensorweb4R"] with 15 slots
## ..@ uom : chr [1:36] "mm" "mm" "mm" "mm" ...
## ..@ phenomenon :Formal class 'Phenomenon' [package "sensorweb4R"] with 5 slots
## ..@ service :Formal class 'Service' [package "sensorweb4R"] with 8 slots
## ..@ feature :Formal class 'Feature' [package "sensorweb4R"] with 5 slots
## ..@ offering :Formal class 'Offering' [package "sensorweb4R"] with 5 slots
## ..@ procedure :Formal class 'Procedure' [package "sensorweb4R"] with 5 slots
## ..@ category :Formal class 'Category' [package "sensorweb4R"] with 4 slots
## ..@ station :Formal class 'Station' [package "sensorweb4R"] with 4 slots
## ..@ firstValue :Formal class 'TVP' [package "sensorweb4R"] with 2 slots
## ..@ lastValue :Formal class 'TVP' [package "sensorweb4R"] with 2 slots
## ..@ referenceValues:List of 36
## ..@ statusIntervals:List of 36
## ..@ id : chr [1:36] "159" "160" "162" "165" ...
## ..@ label : chr [1:36] "Niederschlagshöhe Stundensumme, Barmen Wupperverband Hauptverwaltung
## ..@ endpoint :Formal class 'Endpoint' [package "sensorweb4R"] with 2 slots
```

```
fts <- fetch(ts.precip)
str(fts, 2)
```

```
## Formal class 'Timeseries' [package "sensorweb4R"] with 15 slots
## ..@ uom : chr [1:36] "mm" "mm" "mm" "mm" ...
## ..@ phenomenon :Formal class 'Phenomenon' [package "sensorweb4R"] with 5 slots
## ..@ service :Formal class 'Service' [package "sensorweb4R"] with 8 slots
## ..@ feature :Formal class 'Feature' [package "sensorweb4R"] with 5 slots
## ..@ offering :Formal class 'Offering' [package "sensorweb4R"] with 5 slots
## ..@ procedure :Formal class 'Procedure' [package "sensorweb4R"] with 5 slots
## ..@ category :Formal class 'Category' [package "sensorweb4R"] with 4 slots
## ..@ station :Formal class 'Station' [package "sensorweb4R"] with 4 slots
## ..@ firstValue :Formal class 'TVP' [package "sensorweb4R"] with 2 slots
## ..@ lastValue :Formal class 'TVP' [package "sensorweb4R"] with 2 slots
## ..@ referenceValues:List of 36
## ..@ statusIntervals:List of 36
## ..@ id : chr [1:36] "159" "160" "162" "165" ...
## ..@ label : chr [1:36] "Niederschlagshöhe Stundensumme, Barmen Wupperverband Hauptverwaltung"
## ..@ endpoint :Formal class 'Endpoint' [package "sensorweb4R"] with 2 slots
```

```
time <- as.interval(days(89), ymd("2013-01-01"))
ltvp <- getData(fts, time = time)
ldf <- lapply(ltvp, as.data.frame.TVP)

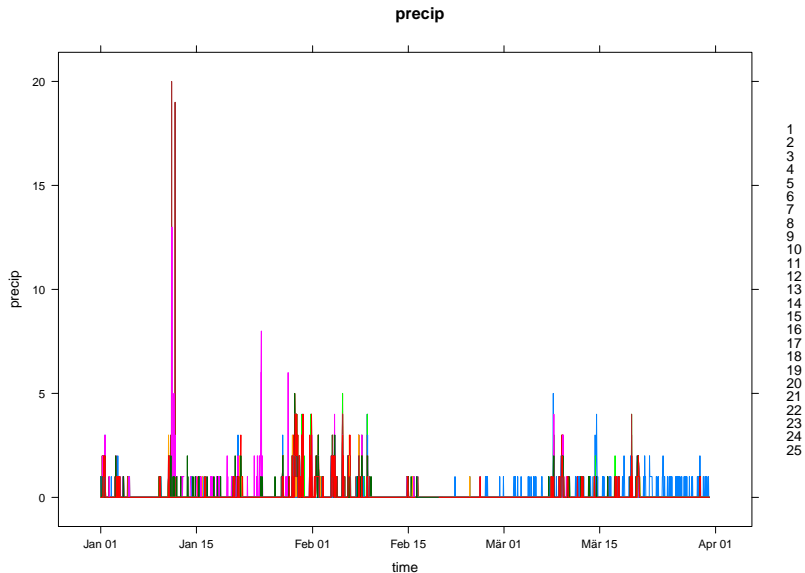
boolPosLength <- sapply(ldf, function(x) dim(x)[1] > 0)

fts <- fts[boolPosLength]
ldf <- ldf[boolPosLength]

mxts <- do.call(merge, lapply(ldf, function(x) xts(x$value, x$time)))
```

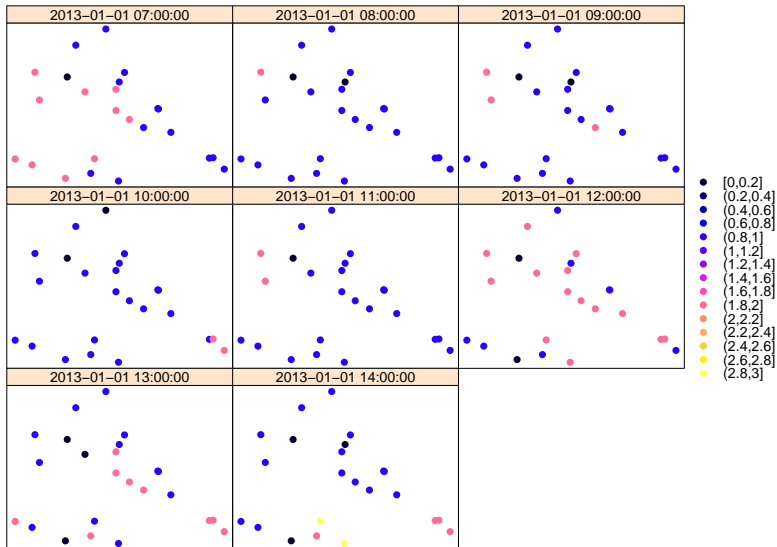


```
stfdf <- STFDF(fts@station@geometry, index(mxts), data.frame(precip=as.numeric(t(mxts))))
stfdf@data$precip[stfdf@data$precip > 300] <- NA
stplot(stfdf, mode="ts")
```

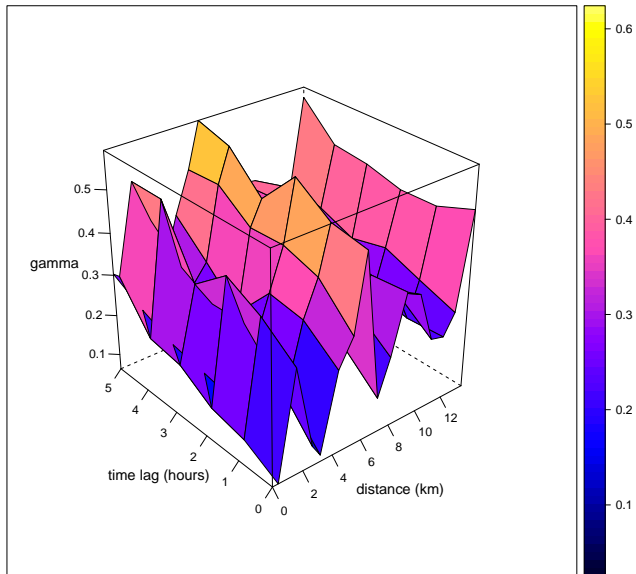


```
library(gstat)
stplot(stfdm[,8:15])
```

precip

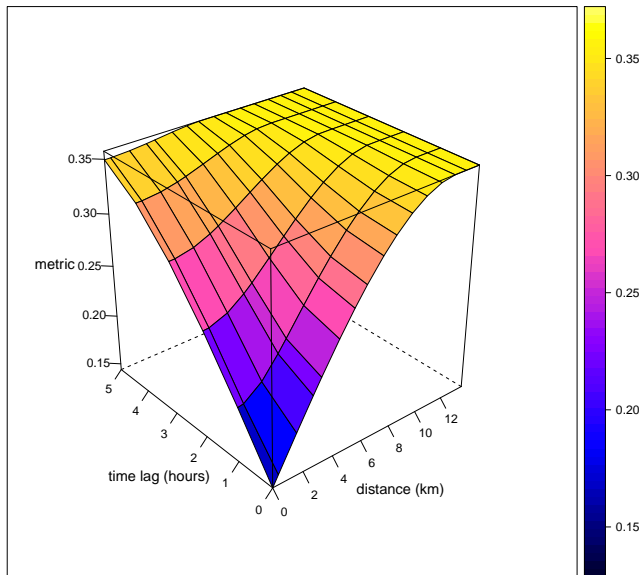


```
empVgm <- variogramST(precip-1, stfdf[,8:15], tlags = 0:5)
plot(empVgm, wireframe=TRUE, scales=list(arrows=FALSE))
```

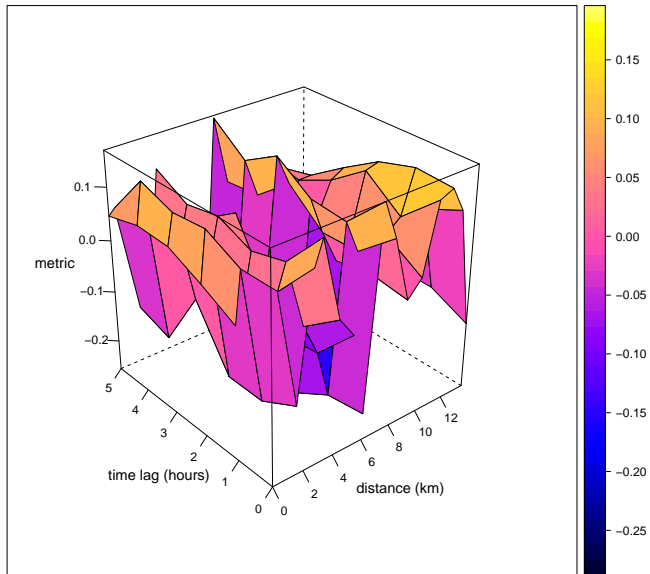


```
fitVgm <- fit.StVariogram(empVgm,  
                           vgmST("metric",  
                                  joint=vgm(0.5,"Sph", 12, 0.1),  
                                  stAni=3))
```

```
plot(empVgm, fitVgm, wireframe=TRUE, scales=list(arrows=FALSE))
```



```
plot(empVgm, fitVgm, diff=TRUE, wireframe=TRUE, scales=list(arrows=FALSE))
```



```
stf <- STF(SpatialGrid(GridTopology(c(7.05,51.075), c(0.05,0.05), c(11,5)), proj4string = stfdf@sp@proj4st  
          stfdf@time[8:15]))
```

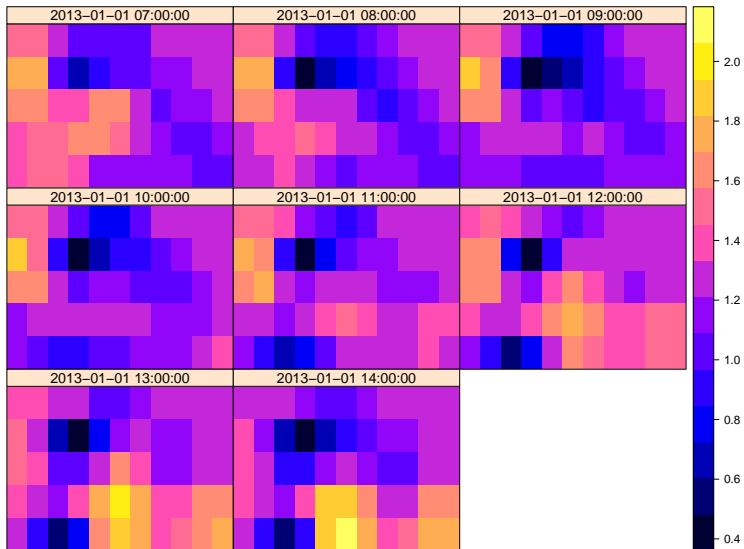
```
## Warning in ST(sp, time, endTime): on constructing ST, converted SpatialGrid  
## to SpatialPixels
```

```
krigedStf <- krigeST(precip-1, as(stfdf[,8:15], "STSDF"), stf, fitVgm)
```

```
## [Using the following time unit: hours]  
## [Using the following time unit: hours]  
## [Using the following time unit: hours]
```

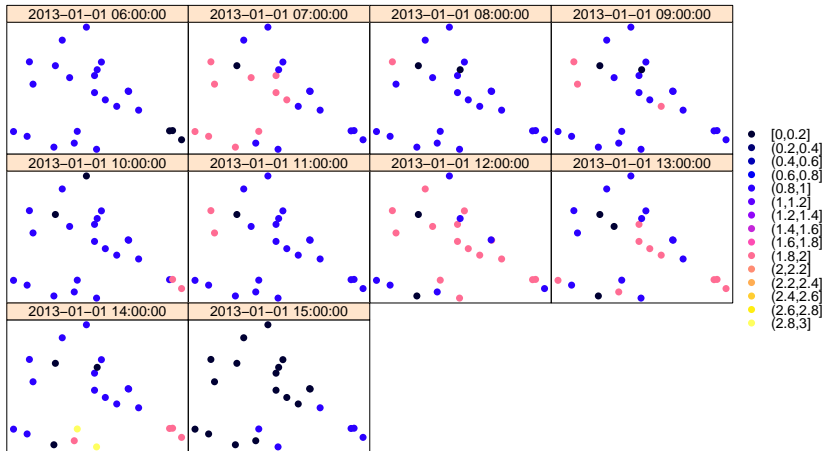
```
stplot(krigedStf)
```

var1.pred




```
stplot(stfdf[,7:16])
```

precip



Shiny

- ▶ R package that generates JS from R function calls
- ▶ also as Shiny server for publication (hosting services available)
- ▶ allows interactive web applications
- ▶ desing separate server logic and ui appearance
- ▶ In: buttons, c/r-boxes, dropdown menus, slider, free text, file upload, ...
- ▶ Out: plots, text, tables (DT: sort- and filterable), file download, ...
- ▶ sensorweb4R + shiny = sensorweby
- ▶ shiny Apps can also be used to make the R-ArcGIS bridge interactive

useful (spatial) R stuff

- ▶ GUI/IDE: RStudio (also RStudio-server)
- ▶ package installation from e.g. GitHub: devtools
- ▶ CRAN packages: sp, spacetime, sf, gstat, rgdal
- ▶ GitHub packages: SOS4R, sensorweb4R, sensorweby
- ▶ Mailinglist: R-sig-Geo
- ▶ Task Views: Spatial, Spatio-Temporal