Summer School – day two

Daniel
Alter Ego

Juli 21, 2017

Contents

List of Tables

List of Figures

Table 1: A table of the cars package

speed	dist
4	2
4	10
7	4
7	22
8	16
9	10

1 aha

Look at Figure 1, it shows the output of a magic function.

```
library(TestPackage)
a <- add_2(seq(1:10))
plot(a)
```

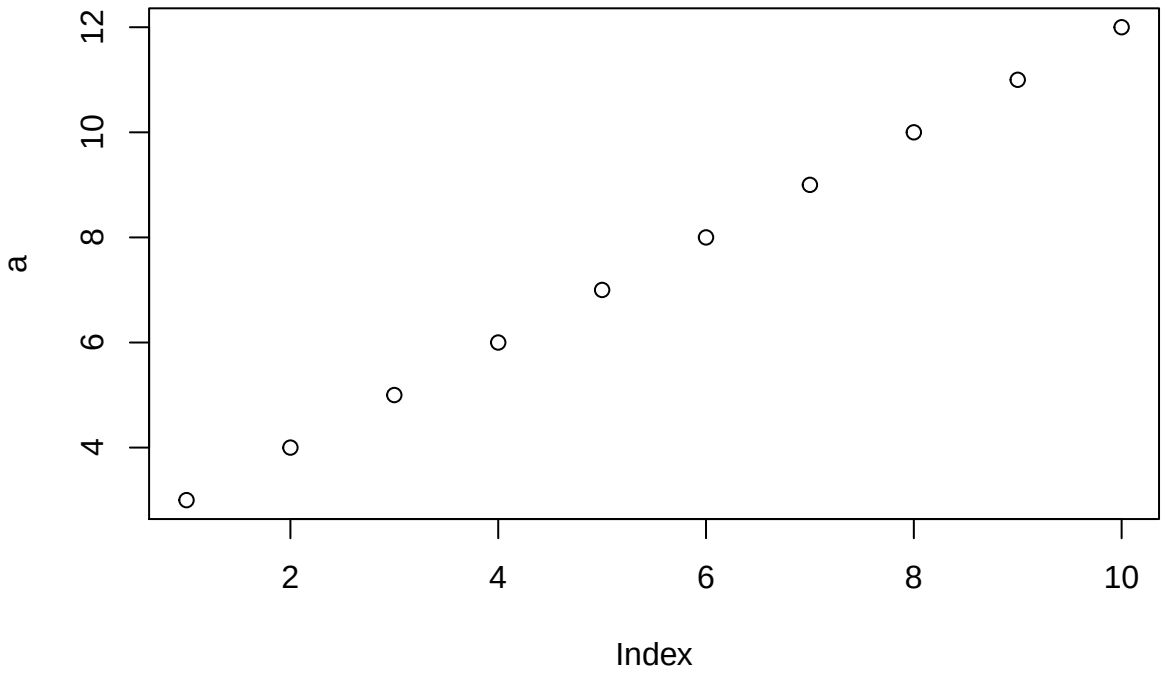


Figure 1: add two

Adding 2 to 4 we get 6.

```
knitr::kable(head(cars),
             caption = "A table of the cars package"
             )
```

Look at the table 1.

Thanks to Xie (2016), Allaire et al. (2017), Xie (2014)

Xie knows what he (?) is doing (2014).

```
add_3(5)
```

```
## [1] 8
```

2 Caching Tests

```
library(tidyverse)
x <- rnorm(1e6) - 5e3
knitr::kable(head(x))
```

3 second column

```
selsecond(mtcars)
```

```
##                     cyl
## Mazda RX4             6
## Mazda RX4 Wag         6
## Datsun 710            4
## Hornet 4 Drive        6
## Hornet Sportabout     8
## Valiant               6
## Duster 360            8
## Merc 240D             4
## Merc 230              4
## Merc 280              6
## Merc 280C             6
## Merc 450SE            8
## Merc 450SL            8
## Merc 450SLC           8
## Cadillac Fleetwood    8
## Lincoln Continental   8
## Chrysler Imperial     8
## Fiat 128              4
## Honda Civic           4
## Toyota Corolla        4
## Toyota Corona         4
## Dodge Challenger      8
## AMC Javelin           8
## Camaro Z28            8
## Pontiac Firebird      8
## Fiat X1-9             4
## Porsche 914-2         4
## Lotus Europa          4
## Ford Pantera L        8
## Ferrari Dino          6
## Maserati Bora         8
## Volvo 142E            4
```

4 Packrat test

```
library(binford)
data(LRB)
knitr::kable(head(LRB))
```

X	seq339	groupno	name	year	ethref
Punan	1	1	Punan_(Borneo)	1970	Kedit 1982 Harrison 1949 Avadhani 1975
Batek	2	2	Batek_Phillipines	1968	Eder 1987 Cadelina 1982
Kubu	3	3	Kubu-(Ridan)	1900	NA
Shompen	4	4	Shompen	1989	Rivzi 1990
Onge	5	5	Onge	1952	Heine-Geldern Hoehn-Gerlachstein 1958 Sen 1962 Cooper 199
Jarwa	6	6	Jarwa	1906	Temple 1903 Radcliffe-Brown 1948

```
harran <- read.table("../data/Sites_HarranPlain.csv", sep = ",", header=TRUE)
str(harran)
```

```
## 'data.frame':    344 obs. of  5 variables:
##  $ X.1            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name           : Factor w/ 166 levels "Ahmet Aslan Tar. (FALSCH)",..: 9 40 40 62 62 62 62 64 64 6
##  $ X              : num  38.8 38.9 38.9 38.9 38.9 ...
##  $ Y              : num  37.6 37.7 37.7 37.2 37.2 ...
##  $ Mentioned_Epoch: Factor w/ 179 levels "","-","Aceramic Neolithic ",..: 175 150 139 162 108 151 16
```

```
library(sp)
coordinates(harran) <- ~X+Y
proj4string(harran) <- CRS("+init=epsg:4326")
str(harran)
```

```
## Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
##   ..@ data       :'data.frame':  344 obs. of  3 variables:
##   .. ..$ X.1            : int [1:344] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ Name           : Factor w/ 166 levels "Ahmet Aslan Tar. (FALSCH)",..: 9 40 40 62 62 62 62 6
##   .. ..$ Mentioned_Epoch: Factor w/ 179 levels "","-","Aceramic Neolithic ",..: 175 150 139 162 108
##   ..@ coords.nrs : int [1:2] 3 4
##   ..@ coords     : num [1:344, 1:2] 38.8 38.9 38.9 38.9 38.9 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:344] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:2] "X" "Y"
##   ..@ bbox       : num [1:2, 1:2] 38.8 36.7 39.2 38.8
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "X" "Y"
##   .. .. ..$ : chr [1:2] "min" "max"
##   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. ..@ projargs: chr "+init=epsg:4326 +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=
```

```
library(raster)
##srtm <- getData("SRTM", lon=38, lat=37)
srtm <- raster("srtm_44_05.tif")
plot(srtm)
points(harran)
```

```
srtm <- crop(srtm, extent(harran)+1)
plot(srtm)
srtm <- projectRaster(srtm, crs = CRS("+init=epsg:32637"))
srtm2 <- aggregate(srtm, fact = 2)
writeRaster(srtm2, "data/dem.tif", overwrite = TRUE)
```

4.1 create point pattern object

```
harran <- spTransform(harran, CRSobj = CRS("+init=epsg:32637"))
library(spatstat)
```

```
## Loading required package: nlme
```

```
## Loading required package: rpart
```

```
##
## spatstat 1.51-0       (nickname: 'Poetic Licence')
## For an introduction to spatstat, type 'beginner'
```

```
##
## Note: spatstat version 1.51-0 is out of date by more than 11 weeks; a newer version should be availa
```

```
harran_ppp <- ppp(x = harran@coords[,1],
              y = harran@coords[,2],
              window = owin(xrange = harran@bbox[1,],
                            yrange = c(min(harran@coords[,2]), min(harran@coords[,2]+52000))))
```

```
## Warning: 65 points were rejected as lying outside the specified window
```

```
## Warning: data contain duplicated points
```

```
plot(harran)
```

4.1.1 challenge: delete duplicated points

```
anyDuplicated(harran_ppp)
```

```
## [1] 2
```

```
harran <- unique(harran_ppp)
plot(harran_ppp)
```

```
## Warning: Interpretation of arguments maxsize and markscale has changed (in
## spatstat version 1.37-0 and later). Size of a circle is now measured by its
## diameter.
```

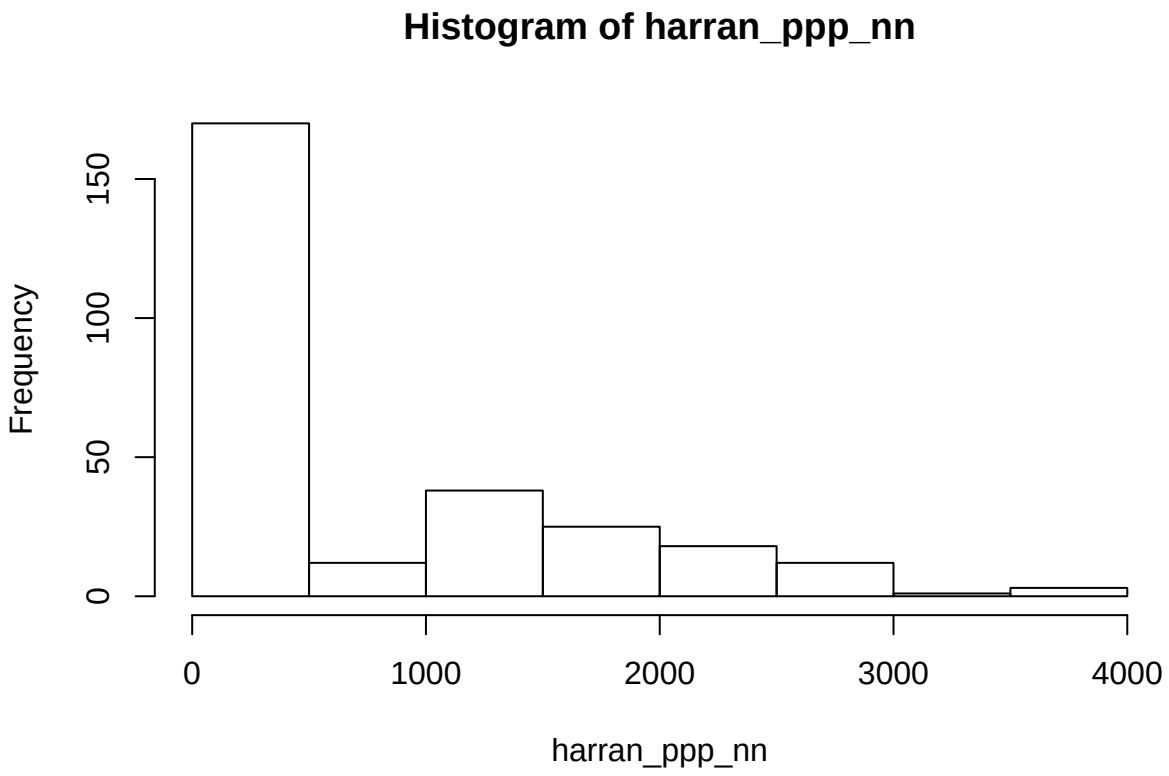
```
## Warning in plot.ppp(harran_ppp): 65 illegal points also plotted
```

harran_ppp

```
harran_ppp_nn <- nndist(harran_ppp)
str(harran_ppp_nn)
```

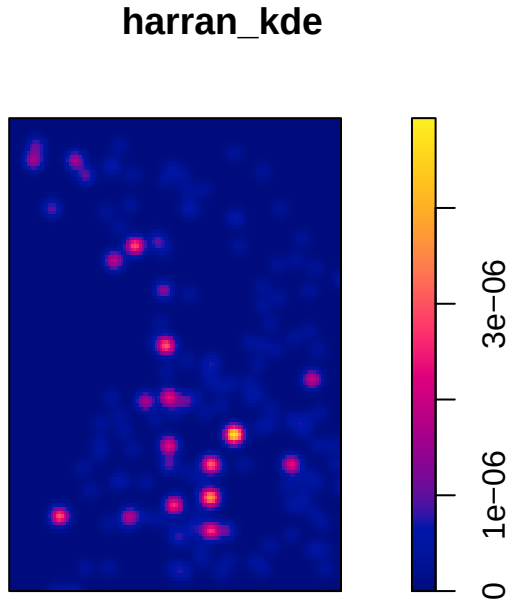
```
##  num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
```

```
hist(harran_ppp_nn)
```



4.2 challenge create kernel density estimation

```
harran_kde <- density.ppp(x = harran_ppp, sigma = mean(harran_ppp_nn))
plot(harran_kde)
```



4.3 raster

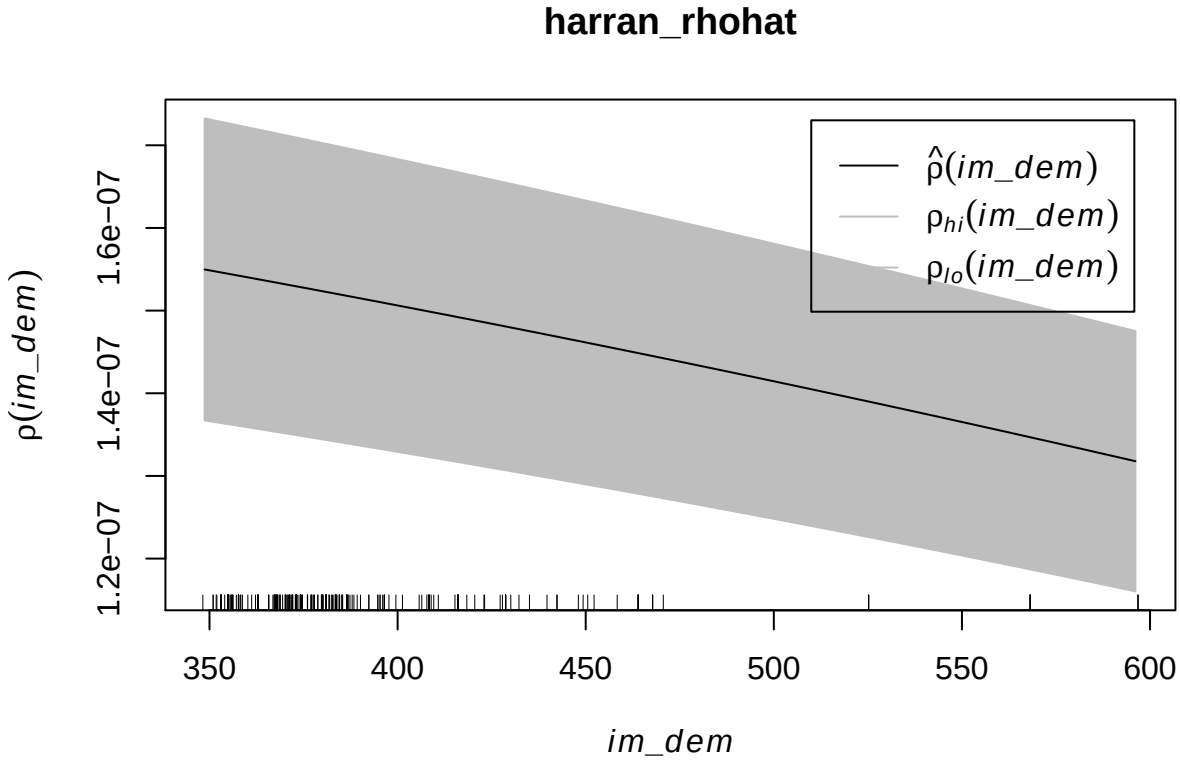
```
library(raster)
```

```
##
## Attaching package: 'raster'
```

```
## The following objects are masked from 'package:spatstat':
##
##     area, rotate, shift
```

```
## The following object is masked from 'package:nlme':
##
##     getData
```

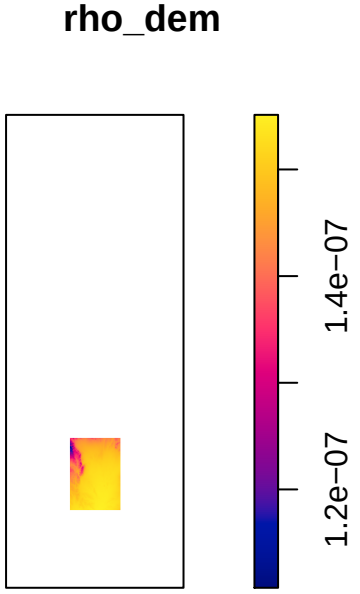
```
dem <- raster("../data/dem.tif")

im_dem <- as.im(as.image.SpatialGridDataFrame(as(dem, "SpatialGridDataFrame")))

harran_rhohat <- rhohat(object = harran_ppp,
                        covariate = im_dem,
                        bw = 200
                        )
plot(harran_rhohat)
```



```
str(harran_rhohat)
```

```
## Classes 'rhohat', 'fv' and 'data.frame': 512 obs. of  5 variables:
##  $ im_dem: num  347 348 349 350 350 ...
##  $ rho   : num  1.55e-07 1.55e-07 1.55e-07 1.55e-07 1.55e-07 ...
##  $ var   : num  8.70e-17 8.69e-17 8.68e-17 8.68e-17 8.67e-17 ...
##  $ hi    : num  1.73e-07 1.73e-07 1.73e-07 1.73e-07 1.73e-07 ...
##  $ lo    : num  1.37e-07 1.37e-07 1.37e-07 1.37e-07 1.37e-07 ...
##  - attr(*, "argu")= chr "im_dem"
##  - attr(*, "valu")= chr "rho"
##  - attr(*, "ylab")= language rho(im_dem)
##  - attr(*, "yexp")= language rho(im_dem)
##  - attr(*, "fmla")= chr ".~im_dem"
##  - attr(*, "alim")= num  348 597
##  - attr(*, "labl")= chr  "im_dem" "hat(%s)(im_dem)" "bold(Var)~hat(%s)(im_dem)" "%s[hi](im_dem)" ...
##  - attr(*, "desc")= chr  "covariate im_dem" "Estimated intensity" "Variance of estimator" "Upper lim
##  - attr(*, "units")=List of 3
##   ..$ singular  : chr "unit"
##   ..$ plural    : chr "units"
##   ..$ multiplier: num 1
##   ..- attr(*, "class")= chr "units"
##  - attr(*, "fname")= chr "rho"
##  - attr(*, "dotnames")= chr  "rho" "hi" "lo"
##  - attr(*, "stuff")=List of 11
##   ..$ modelcall : NULL
##   ..$ callstring: chr "rhohat.ppp(object = harran_ppp, covariate = im_dem, bw = 200)"
##   ..$ sigma     : num 200
##   ..$ covname   : chr "im_dem"
##   ..$ ZX        : num  464 464 371 371 371 ...
##   ..$ lambda    : num  1.47e-07 1.47e-07 1.47e-07 1.47e-07 1.47e-07 ...
##   ..$ method    : chr "ratio"
##   ..$ smoother  : chr "kernel"
##   ..$ reference : chr "Lebesgue"
##   ..$ horvitz   : logi FALSE
##   ..$ Zimage    :List of 10
##   .. ..$ v     : num [1:1842, 1:871] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ dim   : int  1842 871
##   .. ..$ xrange: num  432102 559965
##   .. ..$ yrange: num  4006377 4347147
##   .. ..$ xstep : num 147
##   .. ..$ ystep : num 185
##   .. ..$ xcol  : num  432176 432323 432469 432616 432763 ...
##   .. ..$ yrow  : num  4006469 4006654 4006839 4007024 4007209 ...
##   .. ..$ type  : chr "real"
##   .. ..$ units :List of 3
##   .. .. ..$ singular  : chr "unit"
##   .. .. ..$ plural    : chr "units"
##   .. .. ..$ multiplier: num 1
##   .. .. ..- attr(*, "class")= chr "units"
##   .. ..- attr(*, "class")= chr "im"
```

```
rho_dem <- predict(harran_rhohat)
plot(rho_dem)
```




```
diff_rho <- harran_kde - rho_dem
```

```
## Warning: the images 'e1' and 'e2' were not compatible
```

create random points with rpoispp function that have the same intensity like our point pattern.

```
set.seed(123)
harran_poispp2 <- rpoispp(lambda = harran_ppp$n/area.owin(harran_ppp$window), win = harran_ppp$window)
set.seed(123)
harran_poispp3 <- rpoispp(intensity(harran_ppp), win=Window(harran_ppp))
set.seed(123)
harran_poispp4 <- rpoispp(ex = harran_ppp)

plot(harran_ppp)
```

```
## Warning in plot.ppp(harran_ppp): 65 illegal points also plotted
```

```
points(harran_poispp2, col = "red")
points(harran_poispp3, col = "blue")
points(harran_poispp4, col = "green")
```

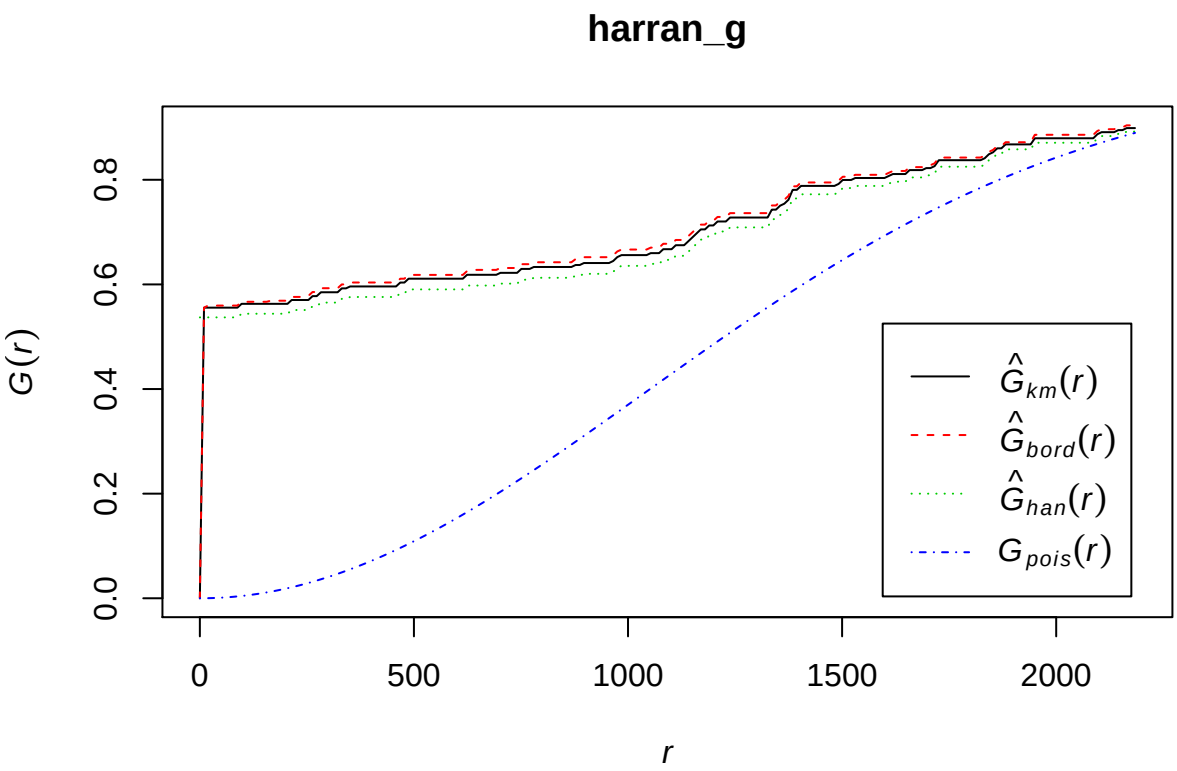
harran_ppp

4.4 Second order effects

```
harran_g <- Gest(harran_ppp)
str(harran_g)
```

```
## Classes 'fv' and 'data.frame':   513 obs. of  7 variables:
##  $ r      : num  0 9.75 19.5 29.25 39 ...
##  $ theo   : num  0.00 4.39e-05 1.76e-04 3.95e-04 7.02e-04 ...
##  $ han    : num  0.537 0.537 0.537 0.537 0.537 ...
##  $ rs     : num  0 0.556 0.56 0.56 0.56 ...
##  $ km     : num  0 0.556 0.556 0.556 0.556 ...
##  $ hazard : num  0 0.0832 0 0 0 ...
##  $ theohaz: num  0.00 9.01e-06 1.80e-05 2.70e-05 3.60e-05 ...
##  - attr(*, "argu")= chr "r"
##  - attr(*, "valu")= chr "km"
##  - attr(*, "ylab")= language G(r)
##  - attr(*, "yexp")= language G(r)
##  - attr(*, "fmla")= chr ".~r"
##  - attr(*, "alim")= num  0 2184
##  - attr(*, "labl")= chr  "r" "%s[pois](r)" "hat(%s)[han](r)" "hat(%s)[bord](r)" ...
##  - attr(*, "desc")= chr  "distance argument r" "theoretical Poisson %s" "Hanisch estimate of %s" "bo
##  - attr(*, "units")=List of 3
##   ..$ singular  : chr "unit"
##   ..$ plural    : chr "units"
##   ..$ multiplier: num 1
##   ..- attr(*, "class")= chr "units"
##  - attr(*, "fname")= chr "G"
##  - attr(*, "dotnames")= chr  "km" "rs" "han" "theo"
```

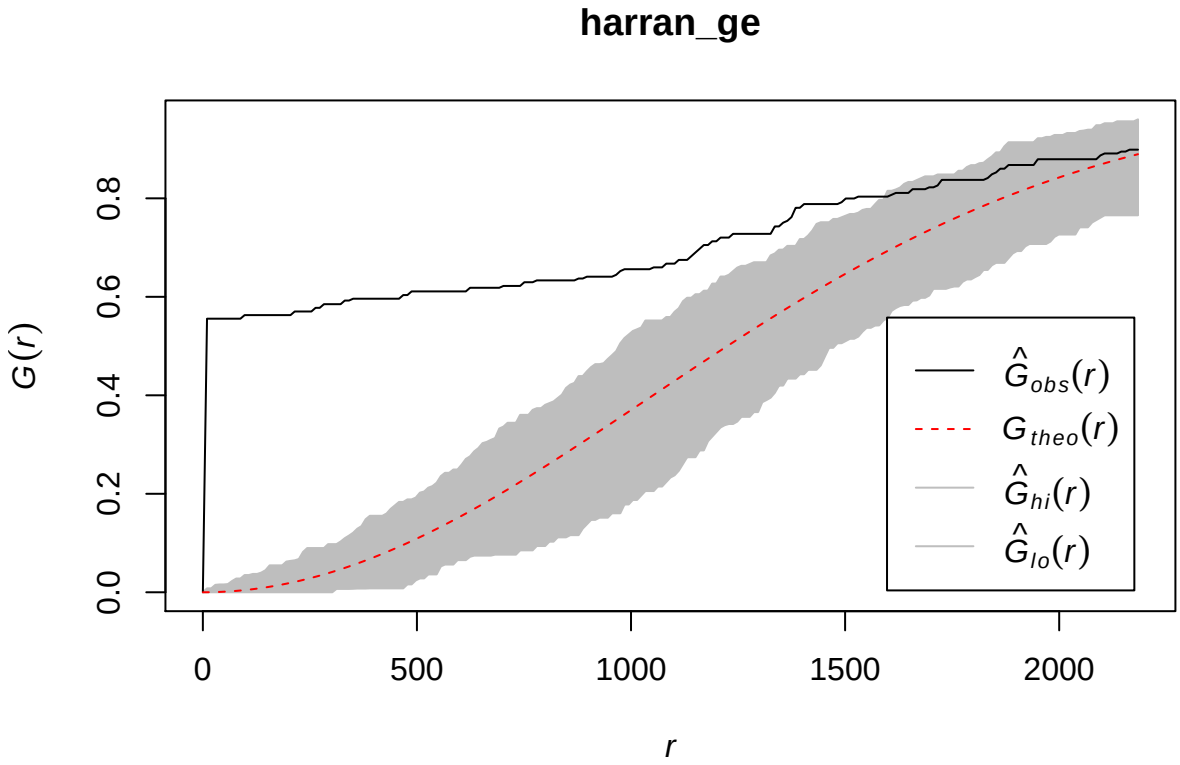
```
plot(harran_g)
```




```
harran_ge <- envelope(harran_ppp, fun = "Gest", nsim = 999)
```

```
## Generating 999 simulations of CSR  ...
## 1, 2, 3, ......10.........20.........30.........40.........50.........60...
## ......70.........80.........90.........100.........110.........120......
## ...130.........140.........150.........160.........170.........180.........
## 190.........200.........210.........220.........230.........240.........250..
## .......260.........270.........280.........290.........300.........310.....
## ....320.........330.........340.........350.........360.........370........
## .380.........390.........400.........410.........420.........430.........440.
## ........450.........460.........470.........480.........490.........500....
## .....510.........520.........530.........540.........550.........560.......
## ..570.........580.........590.........600.........610.........620.........630
## .........640.........650.........660.........670.........680.........690...
## ......700.........710.........720.........730.........740.........750......
## ...760.........770.........780.........790.........800.........810.........
## 820.........830.........840.........850.........860.........870.........880..
## .......890.........900.........910.........920.........930.........940.....
## ....950.........960.........970.........980.........990..... 999.
##
## Done.
```

```
plot(harran_ge)
```



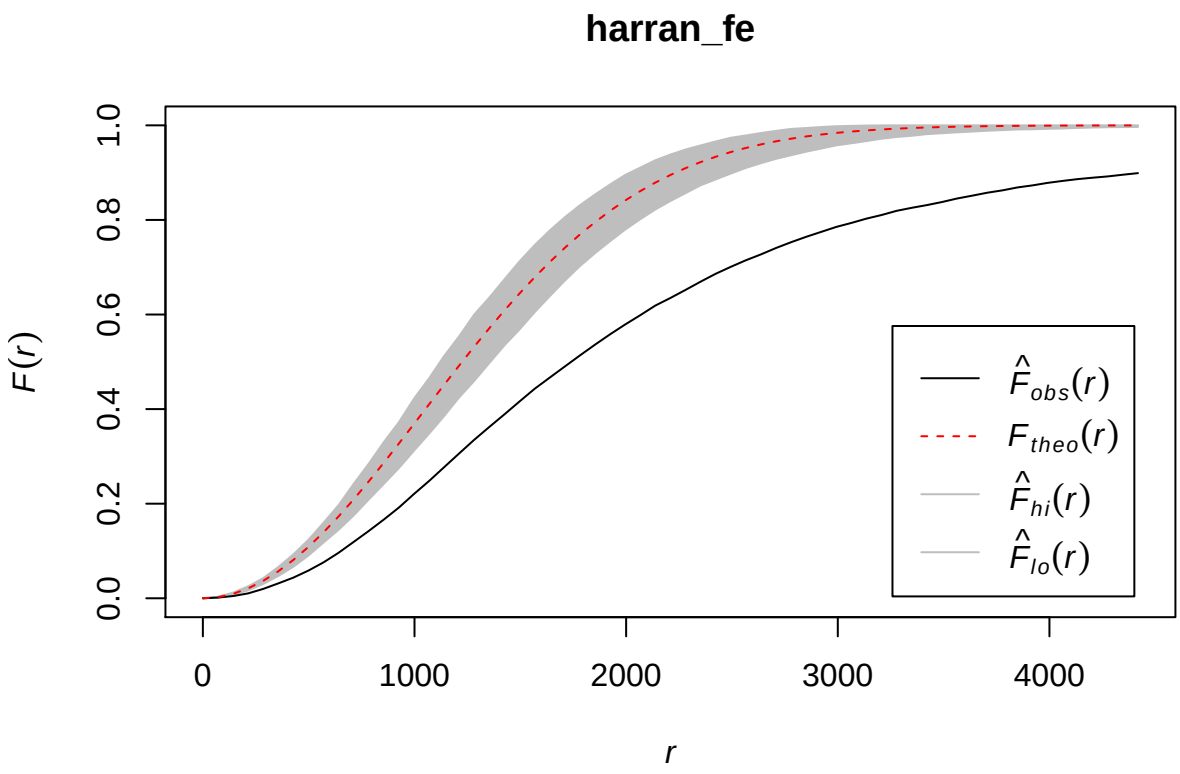

```
harran_f <- Fest(harran_ppp)
harran_fe <- envelope(harran_ppp, fun = "Fest", nsim = 99)
```

```
## Generating 99 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
## 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

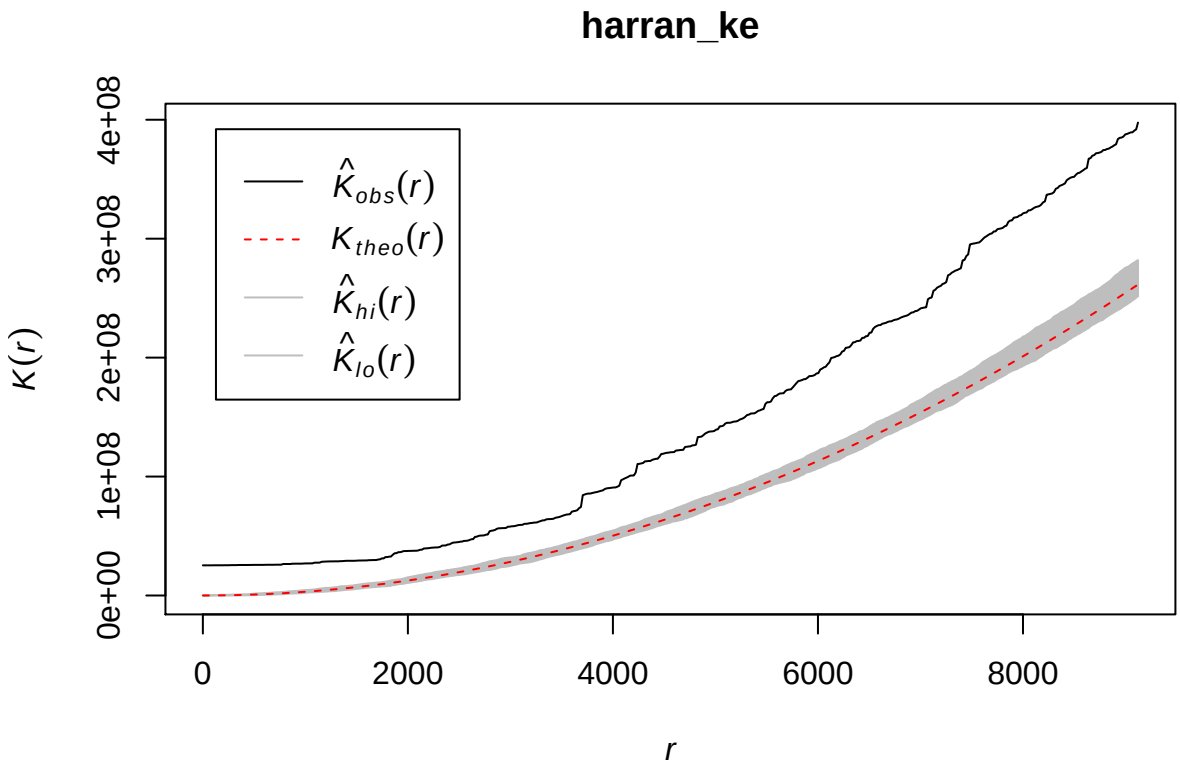
```
harran_k <- Kest(harran_ppp)
harran_ke <- envelope(harran_ppp, fun = "Kest", nsim = 99)
```

```
## Generating 99 simulations of CSR  ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28
## 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 6
## 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,  99.
##
## Done.
```

```
plot(harran_fe)
```




```
plot(harran_ke)
```

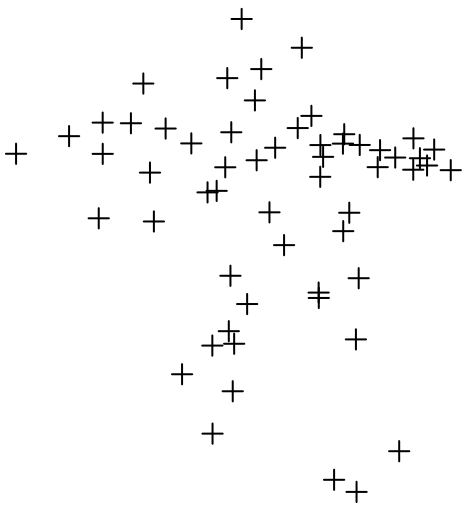


4.5 Inhomogeneous G/F/K

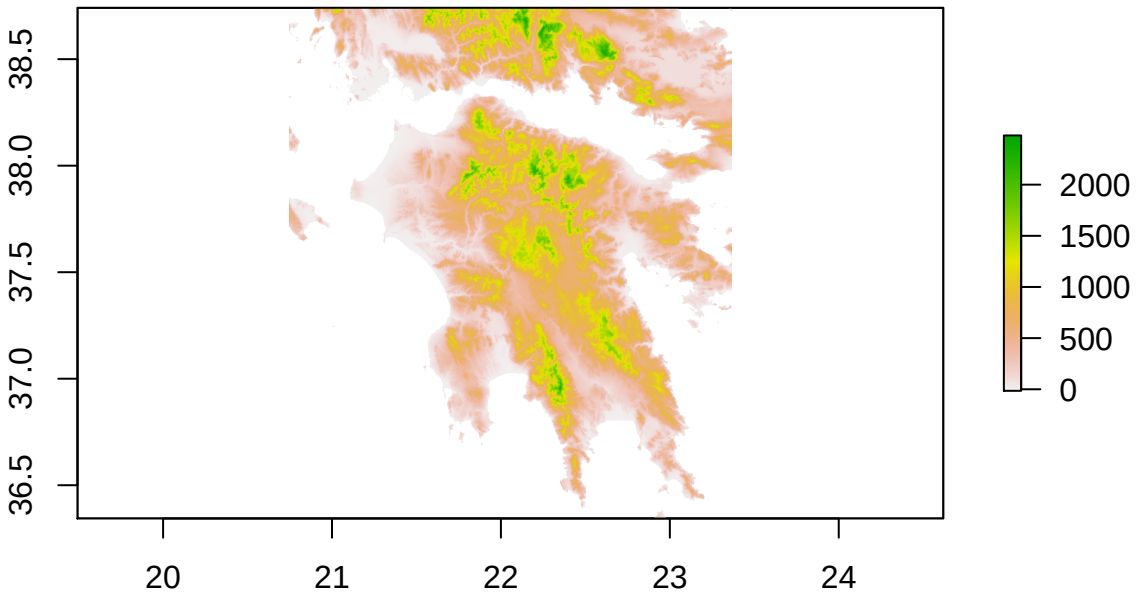
```
harran_gi <- Ginhom(harran_ppp, lambda = predict(harran_rhohat))
#par(mfrow = c(1,2))
#plot(harran_gi, xlim = c(0,6000))
#plot(harran_g, xlim = c(0,6000))
```

5 Interpolation

```
load("../data/Precipitation.RData")
test <- data.frame(test)
library(sp)
coordinates(test) <- ~lon+lat
proj4string(test) <- CRS("+init=epsg:4326")
test2 <- spTransform(test, CRS("+init=epsg:32634"))
plot(test2)
```



```
library(raster)
##srtm <- getData("SRTM", lon=mean(coordinates(test)[,1]), lat=mean(coordinates(test)[,2]))
srtm <- raster("../srtm_41_05.tif")
srtm <- crop(srtm, extent(test)+1)
plot(srtm)
```




```
srtm3 <- projectRaster(srtm, crs=CRS("+init=epsg:32634"))
srtm3 <- aggregate(srtm3, fact = 3)
library(gstat)
```

```
##
## Attaching package: 'gstat'
```

```
## The following object is masked from 'package:spatstat':
##
##     idw
```

5.1 IDW

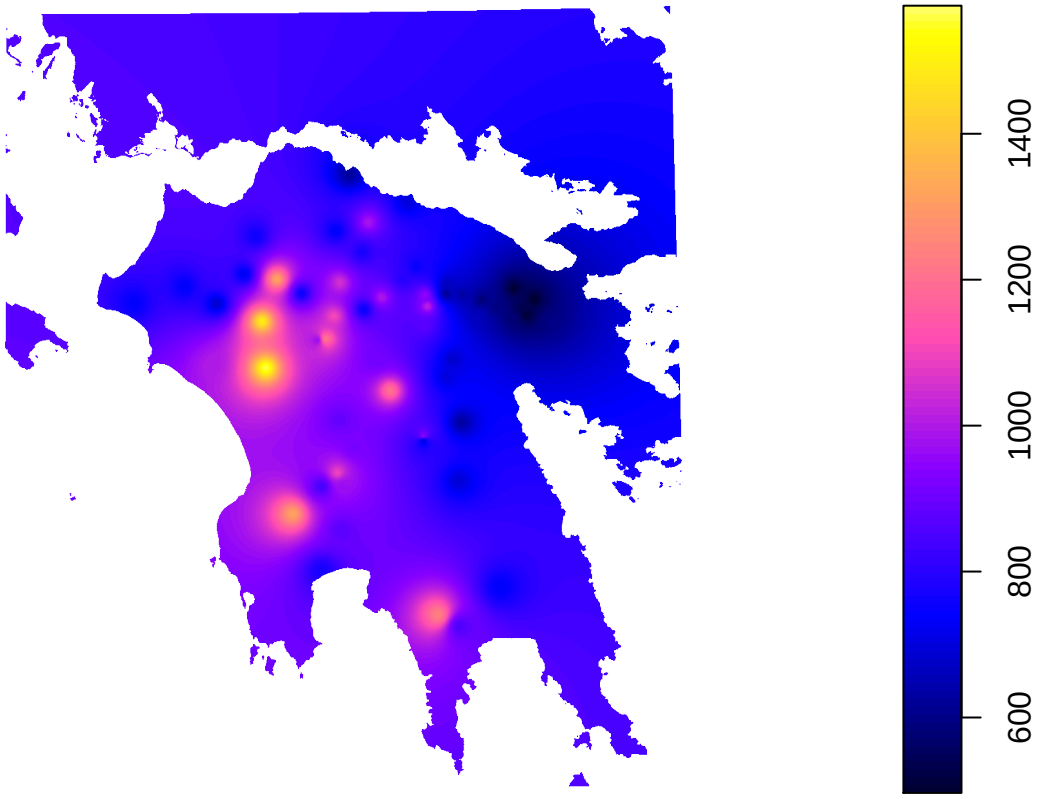
```
rain_idw <- idw(mean_r~1, test2, as(srtm3, "SpatialGridDataFrame"))
```

```
## [inverse distance weighted interpolation]
```

```
rain_idw_cv <- krige.cv(mean_r~1, test2)
head(rain_idw_cv)
```

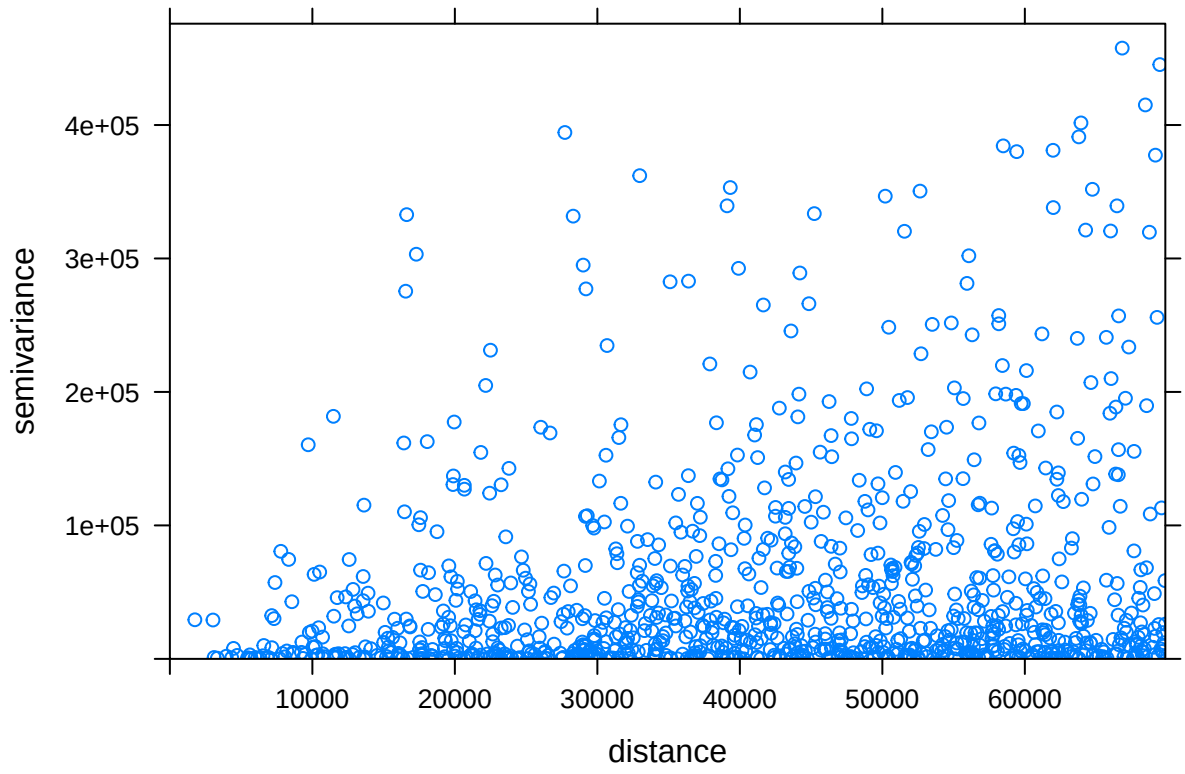
```
##   var1.pred var1.var  observed   residual zscore fold
## 1  568.7307       NA  505.7808  -62.94997     NA    1
## 2  846.9374       NA  626.6029 -220.33449     NA    2
## 3  951.0399       NA  769.8447 -181.19511     NA    3
## 4  823.0326       NA  701.7918 -121.24072     NA    4
## 5  878.1065       NA  775.1778 -102.92874     NA    5
## 6  849.4683       NA 1246.2213  396.75303     NA    6
```

```
plot(rain_idw)
```

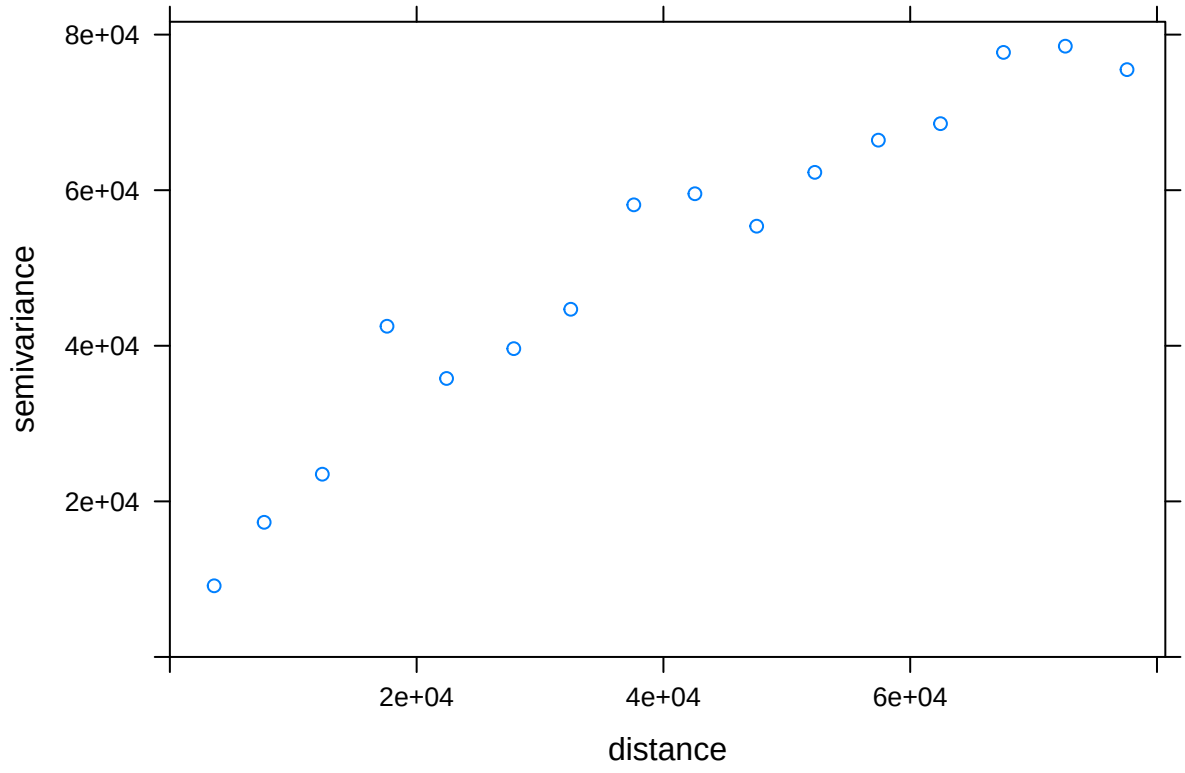


5.2 Kriging

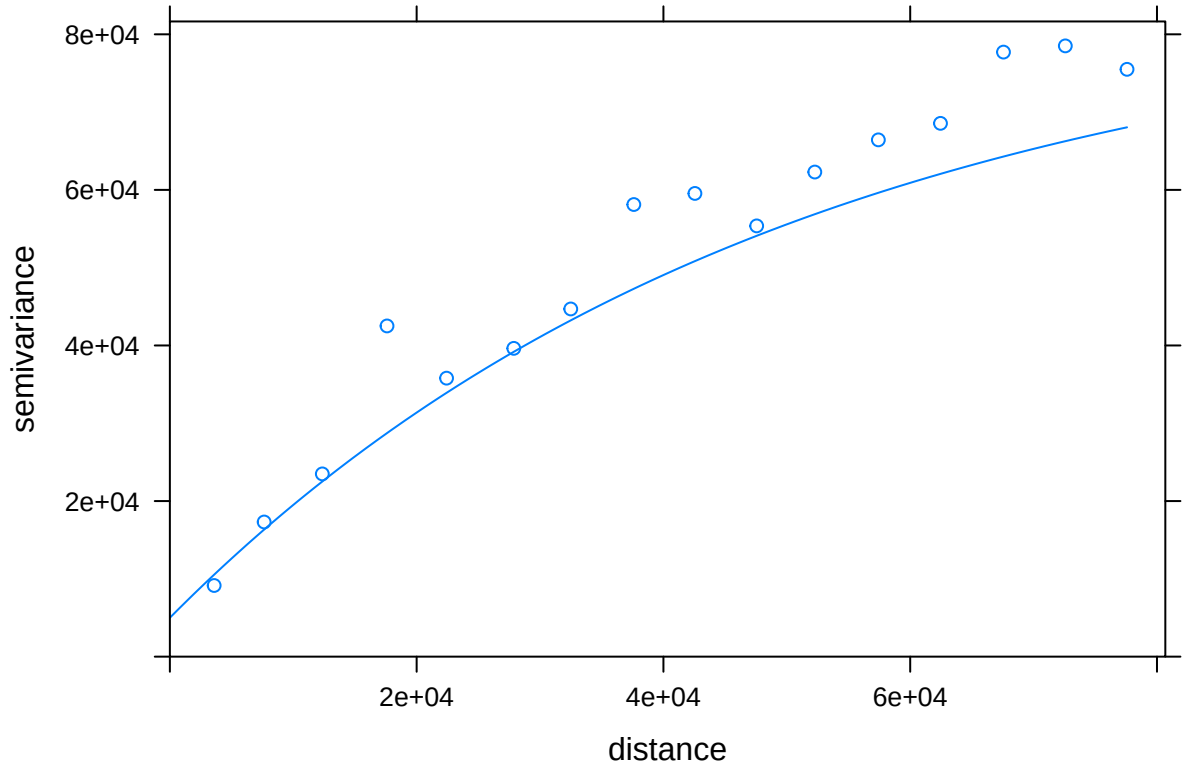
```
plot(variogram(mean_r~1, loc = test2, cloud = TRUE))
```



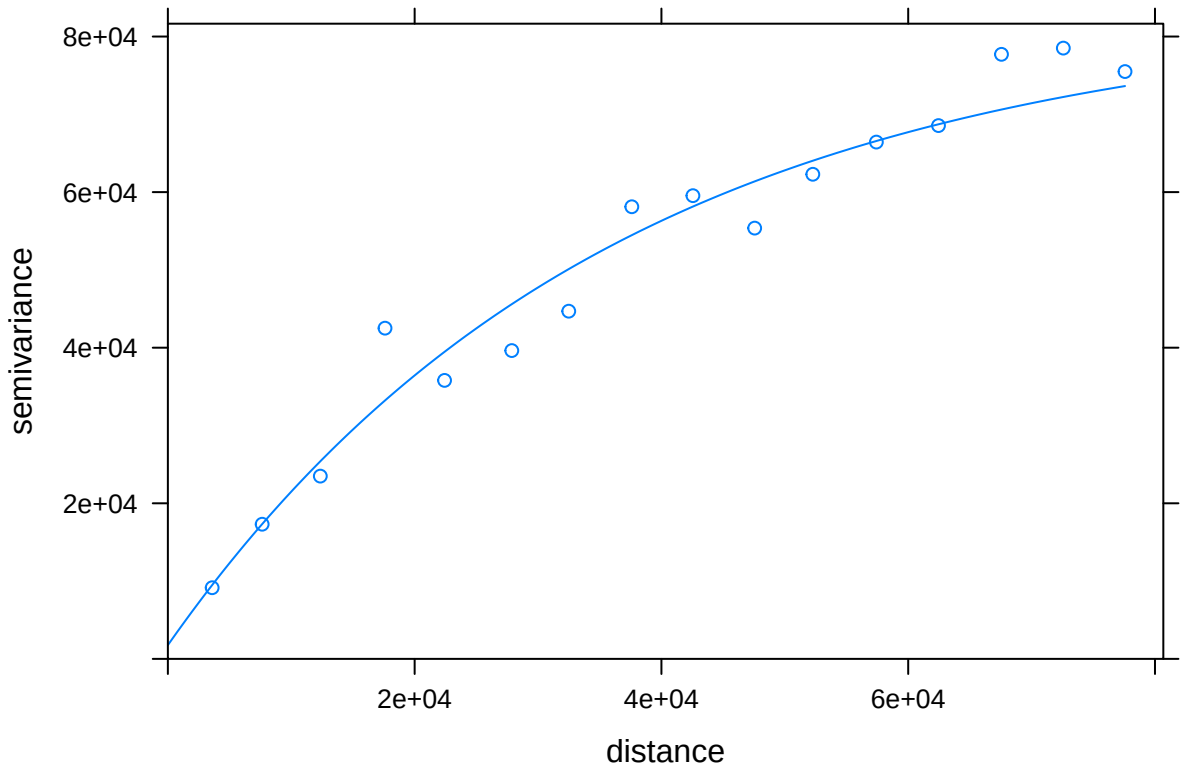

```
va <- variogram(mean_r~1, loc = test2, cutoff = 80000, width = 5000)
plot(va)
```



```
plot(va, vgm(8e+04, "Exp", 5e+04, 5000))
```




```
fva <- fit.variogram(va, vgm(8e+04, "Exp", 5e+04, 5000),
fit.method = 7) ## ordinary least squares; default
plot(va, fva)
```



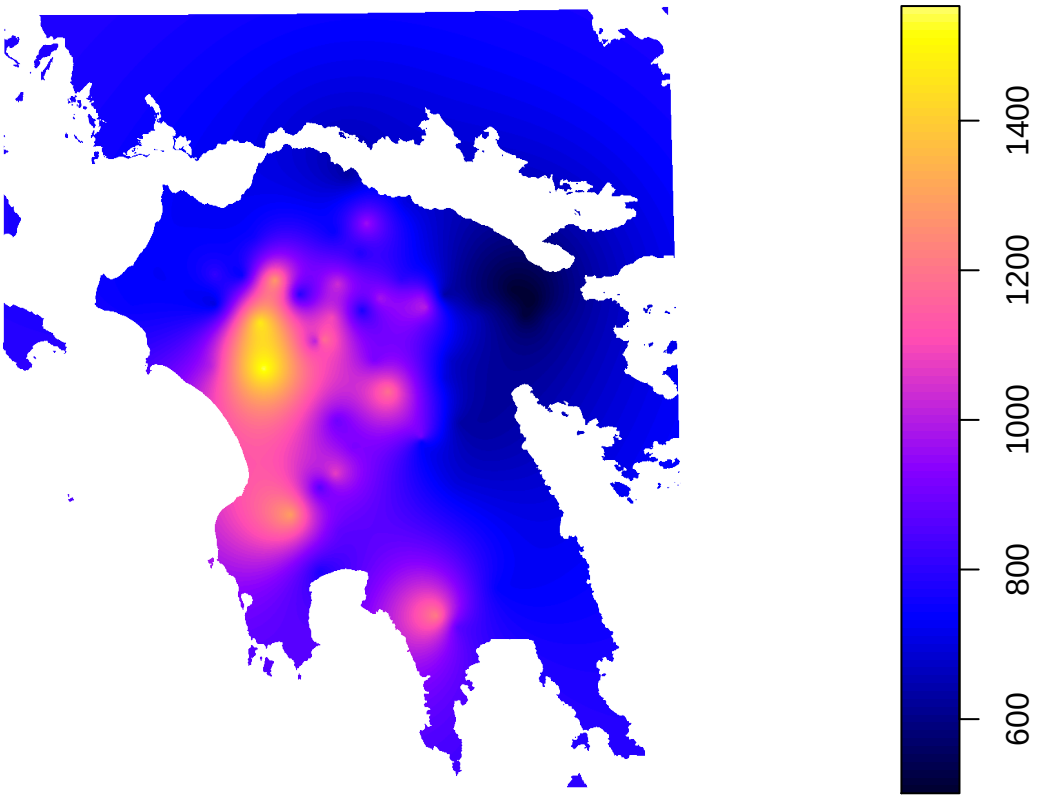

```
rain_krige <- krige(mean_r~1, test2, as(srtm3, "SpatialGridDataFrame"), fva)
```

```
## [using ordinary kriging]
```

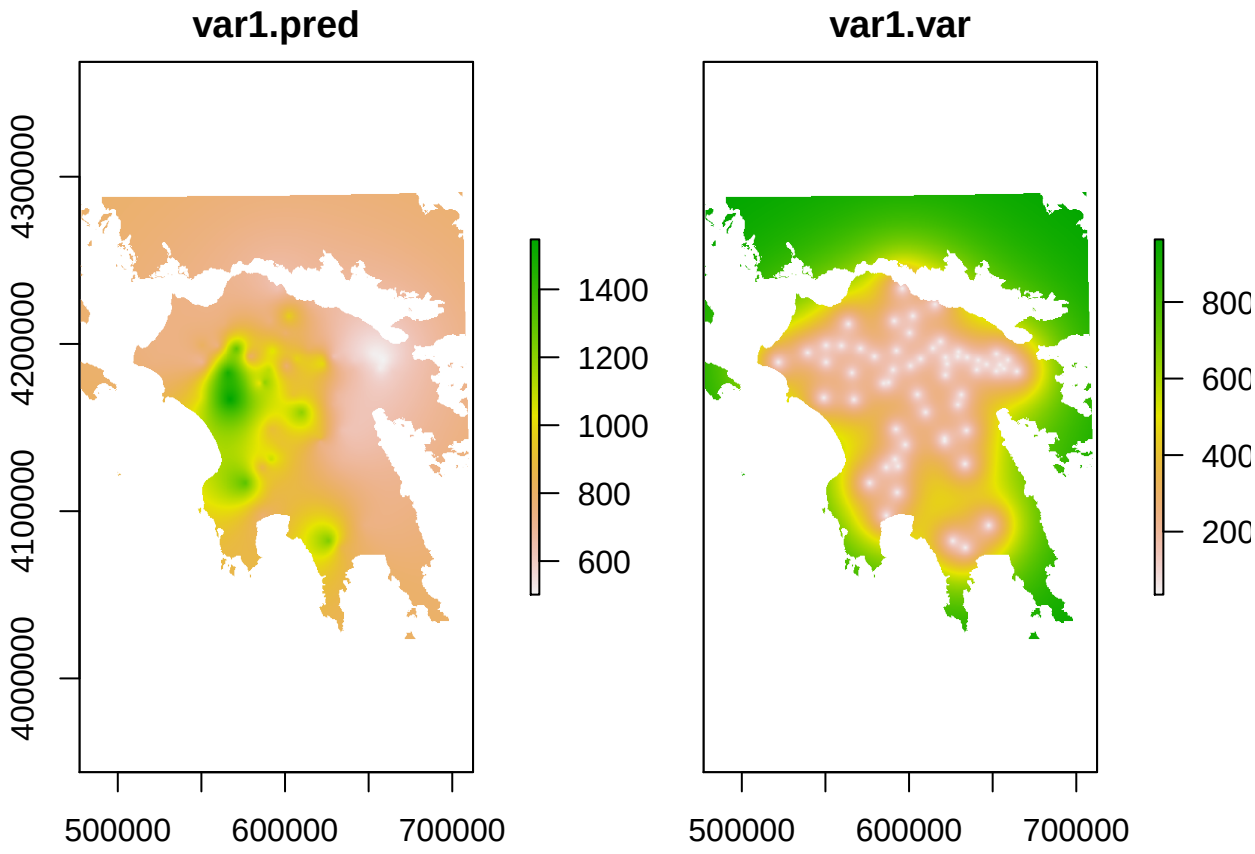
```
rain_krige_cv <- krige.cv(mean_r~1, test2, fva)
head(rain_krige_cv)
```

```
##   var1.pred var1.var  observed  residual     zscore fold
## 1  552.0460 11517.83  505.7808  -46.2652 -0.4310912    1
## 2  804.4267 46791.59  626.6029 -177.8238 -0.8220647    2
## 3 1020.0006 45572.54  769.8447 -250.1558 -1.1718148    3
## 4  787.6132 41457.87  701.7918  -85.8214 -0.4214947    4
## 5  917.3365 23815.72  775.1778 -142.1587 -0.9211739    5
## 6  825.8648 31103.98 1246.2213  420.3565  2.3834701    6
```

```
plot(rain_krige)
```




```
rain_krige2 <- brick(rain_krige)
plot(rain_krige2)
```



5.3 Automatic Kriging

```
library(automap)
rainautkrige <- automap::autoKrige(mean_r~1, test2, as(srtm3, "SpatialGridDataFrame"))
```

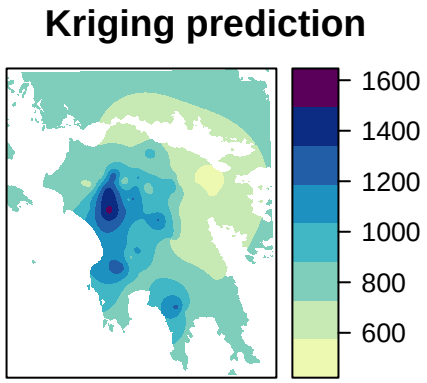
```
## [using ordinary kriging]
```

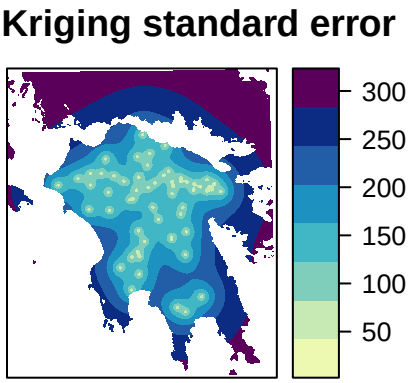
```
str(rainautkrige)
```

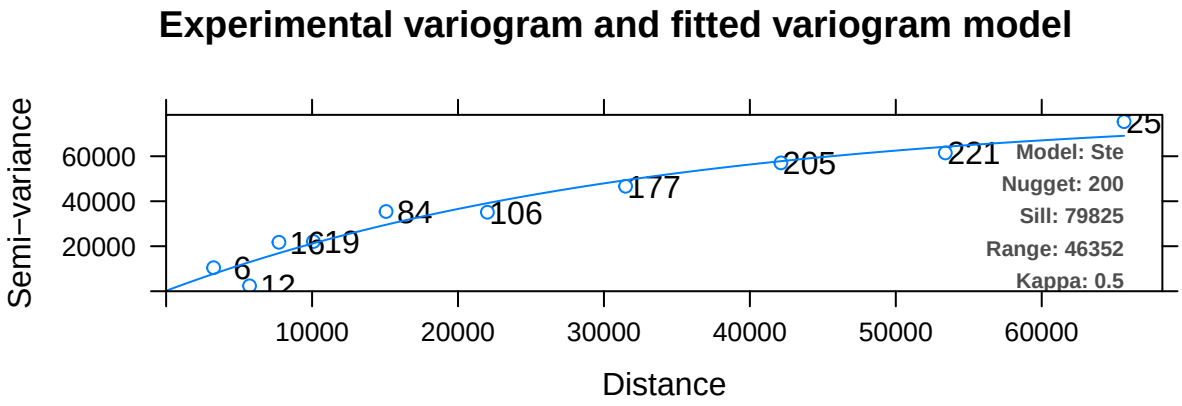
```
## List of 4
##  $ krige_output:Formal class 'SpatialGridDataFrame' [package "sp"] with 4 slots
##   .. ..@ data       :'data.frame':   1036152 obs. of  3 variables:
##   .. .. ..$ var1.pred : num [1:1036152] NA NA NA NA NA NA NA NA NA NA ...
##   .. .. ..$ var1.var  : num [1:1036152] NA NA NA NA NA NA NA NA NA NA ...
##   .. .. ..$ var1.stdev: num [1:1036152] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..@ grid       :Formal class 'GridTopology' [package "sp"] with 3 slots
##   .. .. .. ..@ cellcentre.offset: Named num [1:2] 477198 4021586
##   .. .. .. .. ..- attr(*, "names")= chr [1:2] "s1" "s2"
##   .. .. .. ..@ cellsize         : num [1:2] 221 278
##   .. .. .. ..@ cells.dim        : int [1:2] 1066 972
##   .. ..@ bbox       : num [1:2, 1:2] 477087 4021447 712460 4291177
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "s1" "s2"
```

```
##   .. .. .. ..$ : chr [1:2] "min" "max"
##   .. ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
##   .. .. .. ..@ projargs: chr "+init=epsg:32634 +proj=utm +zone=34 +datum=WGS84 +units=m +no_defs +el
##  $ exp_var     :Classes 'gstatVariogram' and 'data.frame':   10 obs. of  6 variables:
##   ..$ np     : num [1:10] 6 12 16 19 84 106 177 205 221 256
##   ..$ dist   : num [1:10] 3253 5715 7730 10090 15071 ...
##   ..$ gamma  : num [1:10] 10456 2380 21764 22094 35433 ...
##   ..$ dir.hor: num [1:10] 0 0 0 0 0 0 0 0 0 0
##   ..$ dir.ver: num [1:10] 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 [1:10] 4410 6615 8820 11025 18374 ...
##   ..- attr(*, "pseudo")= num 0
##   ..- attr(*, "what")= chr "semivariance"
##  $ var_model   :Classes 'variogramModel' and 'data.frame':   2 obs. of  9 variables:
##   ..$ model: Factor w/ 20 levels "Nug","Exp","Sph",..: 1 7
##   ..$ psill: num [1:2] 200 79625
##   ..$ range: num [1:2] 0 46352
##   ..$ kappa: num [1:2] 0 0.5
##   ..$ ang1 : num [1:2] 0 0
##   ..$ ang2 : num [1:2] 0 0
##   ..$ ang3 : num [1:2] 0 0
##   ..$ anis1: num [1:2] 1 1
##   ..$ anis2: num [1:2] 1 1
##   ..- attr(*, "singular")= logi FALSE
##   ..- attr(*, "SSErr")= num 72.3
##   ..- attr(*, "call")= language fit.variogram(object = experimental_variogram, model = vgm(psill = ps
##  $ sserr       : num 72.3
##  - attr(*, "class")= chr [1:2] "autoKrige" "list"
```

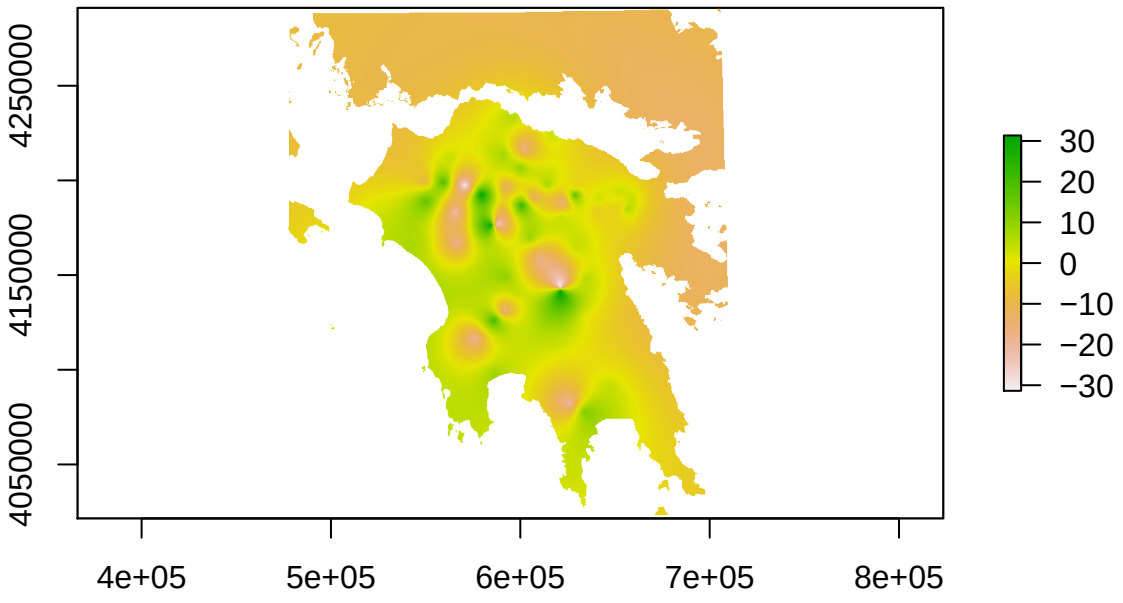
```
plot(rainautkrige)
```








```
tmp <- rainautkrige[[1]]
tmp <- raster(tmp)
tmp2 <- raster(rain_krige)
tmp3 <- tmp2-tmp
plot(tmp3)
## or using Reduce function (that needs a list)
plot(Reduce("-", list(tmp2,tmp)))
```

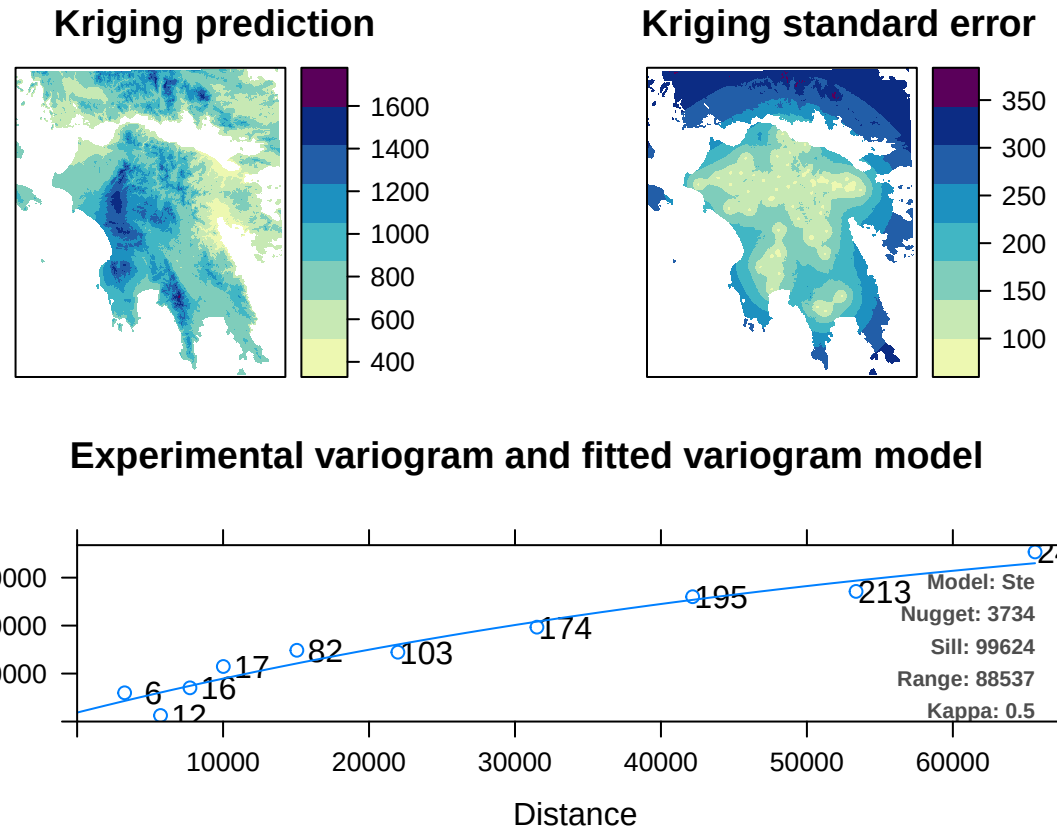


5.4 Kriging with external drift

```
names(srtm3) <- "altitude"
test3 <- test2[!is.na(test2$altitude),]
rainkwed <- automap::autoKrige(mean_r~altitude, test3, as(srtm3, "SpatialGridDataFrame"))
```

```
## [using universal kriging]
```

```
plot(rainkwed)
```



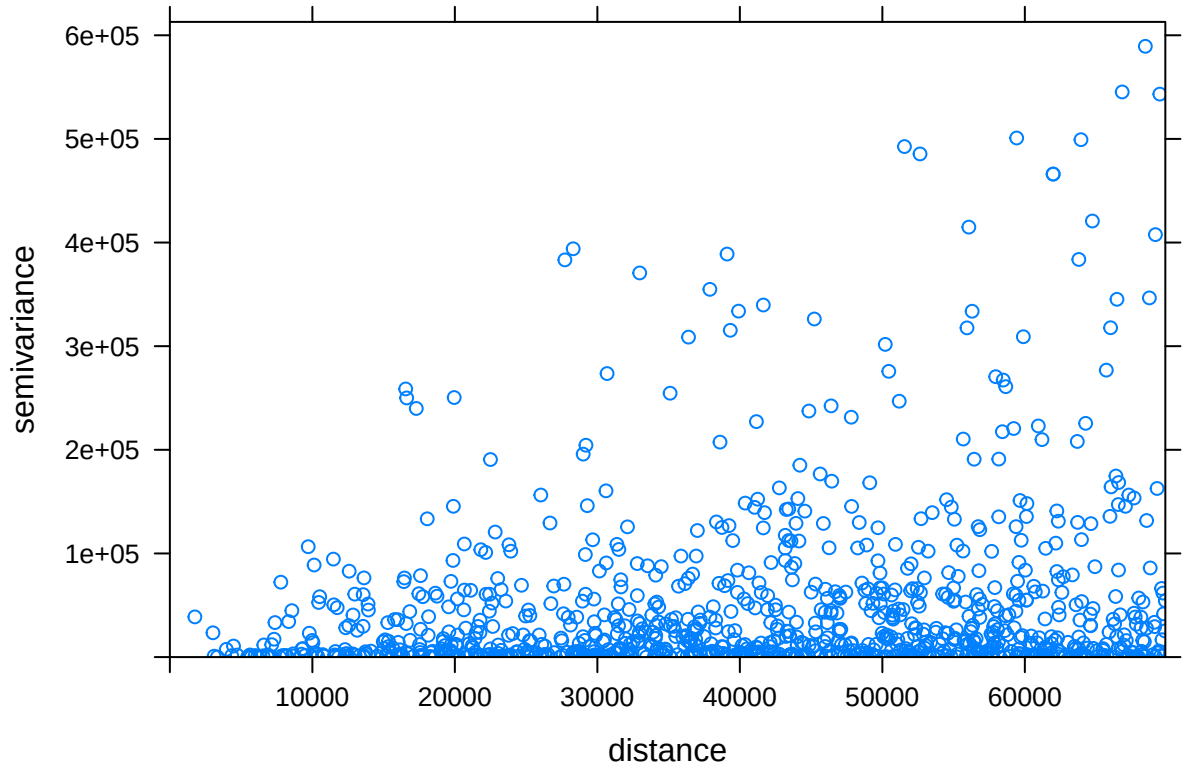
5.5 Kriging with external drift (manual)

see Wolfgang's script.

```
linmod <- lm(mean_r~altitude, test3)
linmod
```

```
##
## Call:
## lm(formula = mean_r ~ altitude, data = test3)
##
## Coefficients:
## (Intercept)     altitude
##    724.2304       0.2337
```

```
plot(variogram(linmod$residuals~1, loc = test3, cloud = TRUE))
```



5.6 the easy way: altitude + x

see Wolfgang's script.

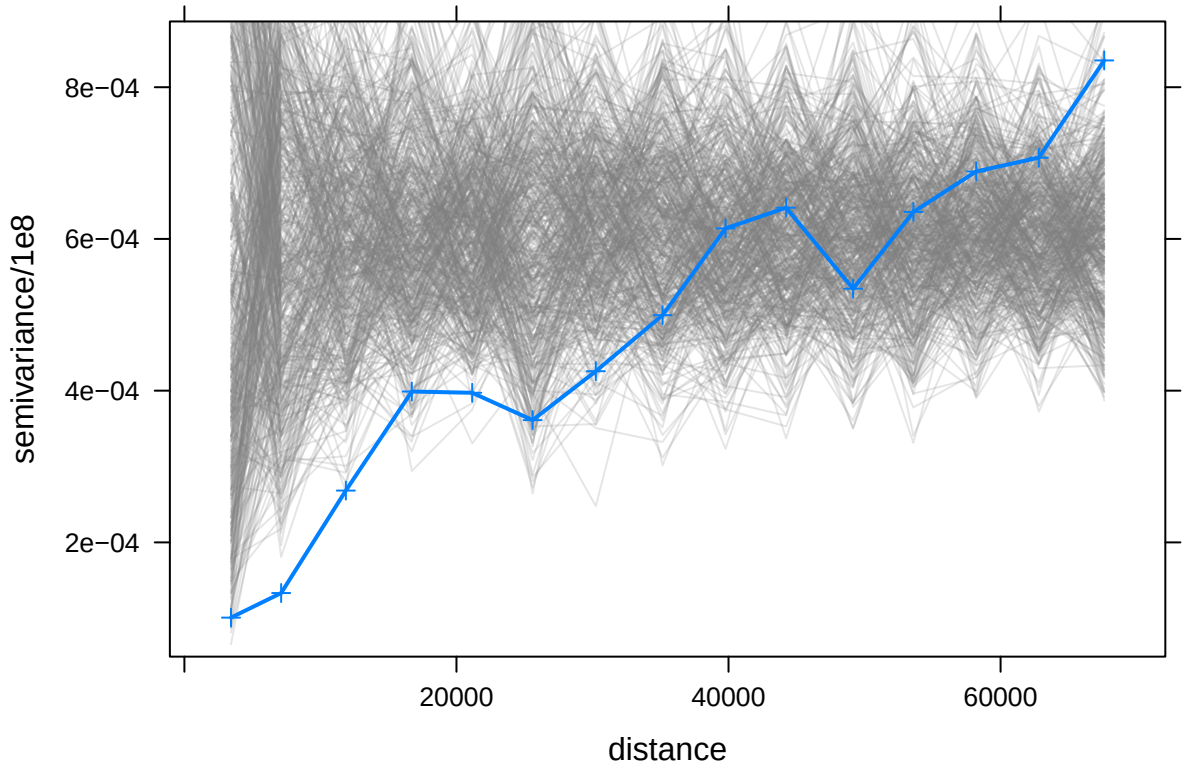
```
test3$x <- coordinates(test3)[,1]
```

```
library(lattice); library(gstat)
```

```
##
## Attaching package: 'lattice'

## The following object is masked from 'package:spatstat':
##
##     panel.histogram
```

```
v <- variogram(mean_r ~ 1, test3)
xyplot(
    x = gamma/1e8 ~ dist,
    data = v,
    pch = 3,
    type = 'b',
    lwd = 2,
    panel = function(x, y, ...) {
        for (i in 1:500) {
```

```
            test3$random = sample(test3$mean_r)
            v = variogram(random ~ 1, test3)
            llines(x = v$dist,
                   y = v$gamma/1e8,
                   col = rgb(.5,.5,.5,.2)
                   )
        }
        panel.xyplot(x, y, ...)
    },
    xlab = 'distance',
    ylab = 'semivariance/1e8'
)
```



References

Allaire, JJ, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, Aron Atkins, Rob Hyndman, and Ruben Arslan. 2017. *Rmarkdown: Dynamic Documents for R*. https://CRAN.R-project.org/package=rmarkdown.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2016. *Bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman; Hall/CRC. https://github.com/rstudio/bookdown.