# Estimation of the variogram and MLE of the parameters in a spatial regression model

Alexandra M. Schmidt

May 2018

This document shows how to perform exploratory data analysis of spatial data using `geoR`. It also shows how to obtain estimates of the empirical variogram, and estimates based on weighted least squares. Finally, we obtain the estimates of the parameters of a spatial regression model for some artificial data.

## 1 Exploratory analysis of spatially structured data

```
library(geoR)



# Loading data:
data(s100)
str(s100)

## List of 7
##  $ NULL: num [1:100, 1:2] 0.8071 0.55 0.3408 0.1371 0.0442 ...
##  $ data                                               : num [1:100] 0.917 1.148 1.033 0.12
##  $ cov.model                                          , nugget
##  $ NA: num 0
##  $ NA: num [1:2] 1 0.3
##  $ NA: num 0.5
##  $ NA: num 1
##  - attr(*, "class")= chr "geodata"

#non-spatial summaries
summary(s100)

## Number of data points: 100
##
## Coordinates summary
```

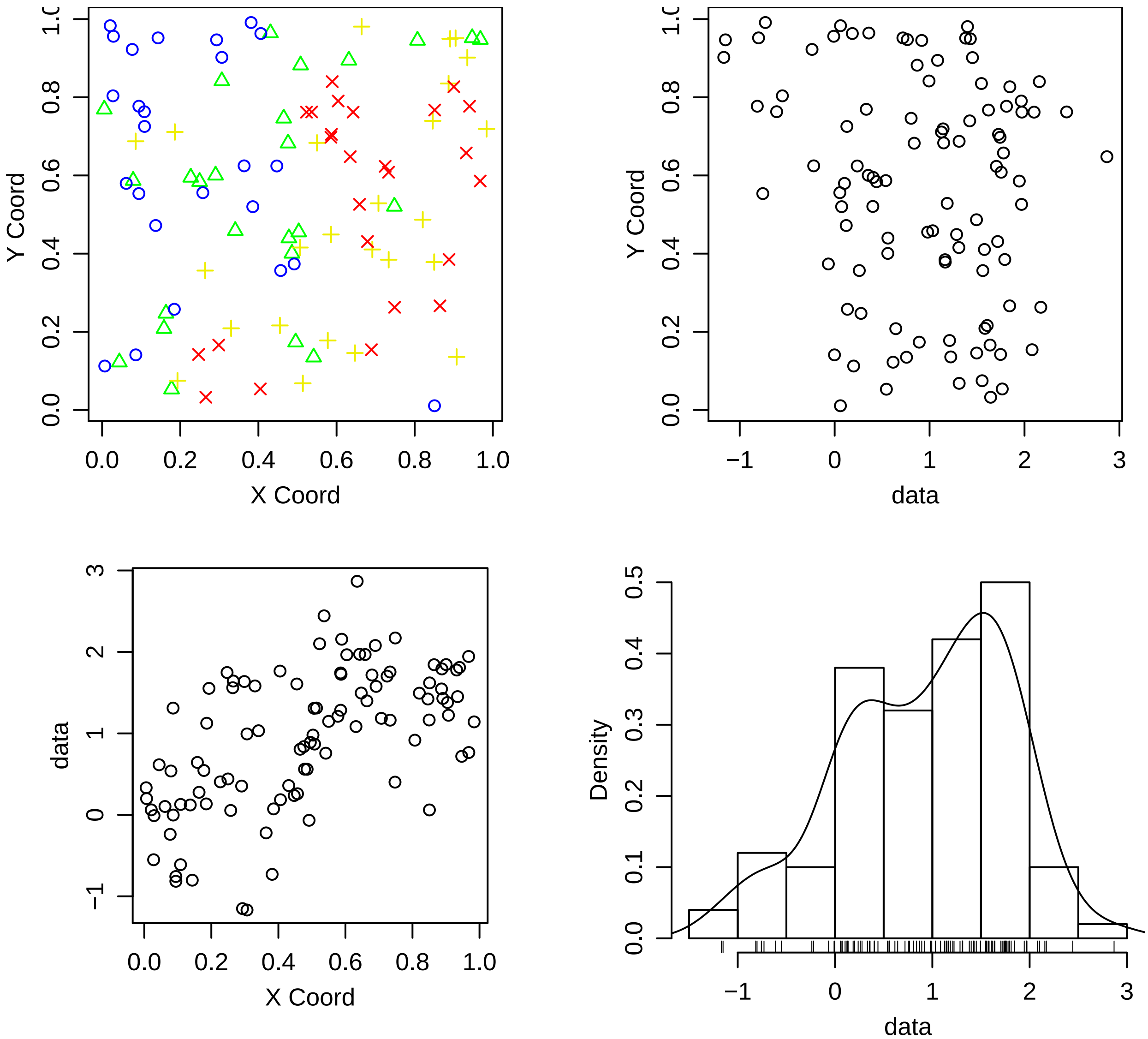
```
##         Coord.X    Coord.Y
## min 0.005638006 0.01091027
## max 0.983920544 0.99124979
##
## Distance summary
##         min         max
## 0.007640962 1.278175109
##
## Data summary
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.1676955  0.2729882  1.1045936  0.9307179  1.6101707  2.8678969
##
## Other elements in the geodata object
## [1] "cov.model" "nugget"    "cov.pars"  "kappa"     "lambda"
```

```
#stem plots
stem(s100$data)
```

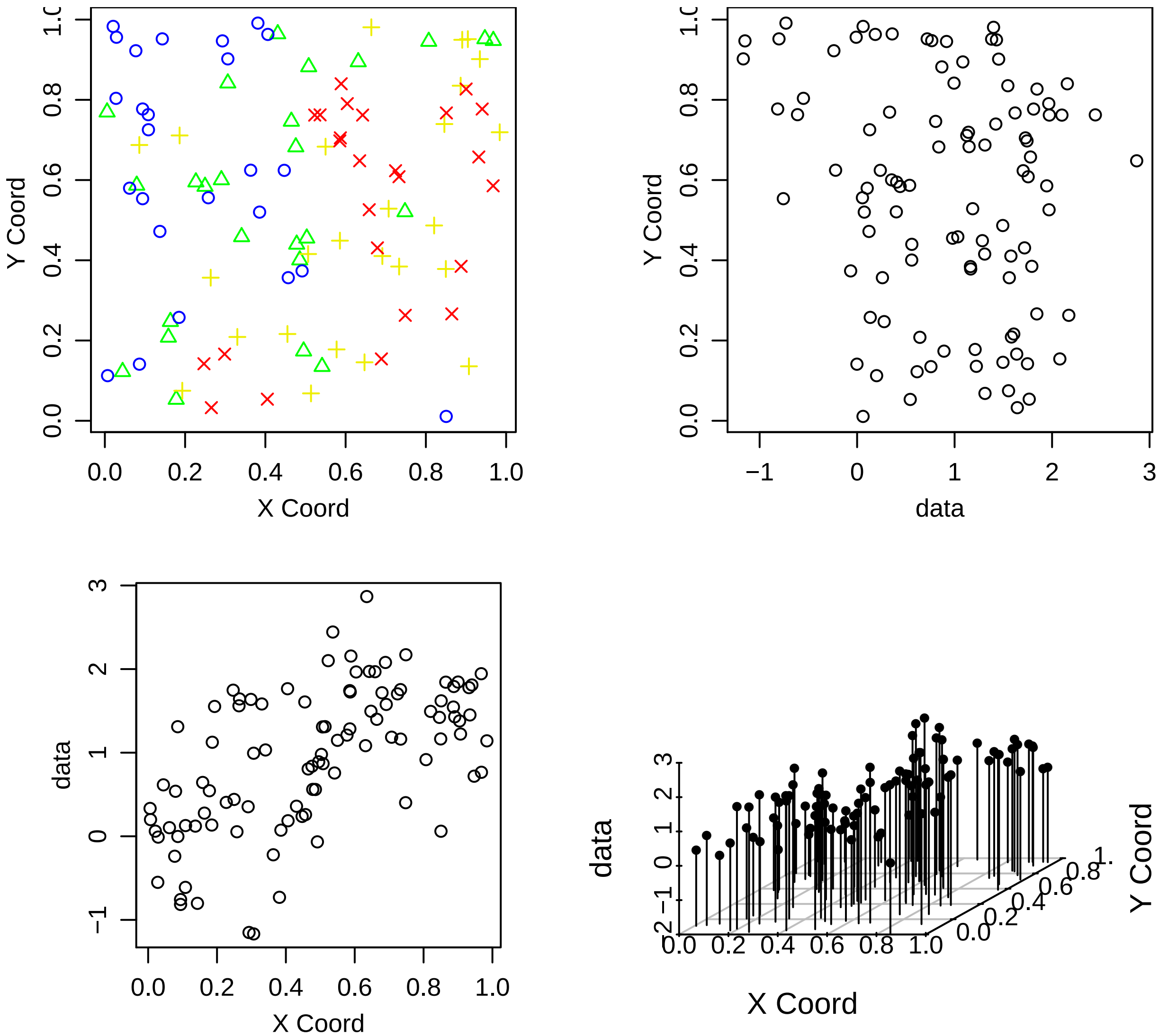
```
##
##   The decimal point is at the |
##
##   -1 | 22
##   -0 | 888766
##   -0 | 22100
##    0 | 1111111122233344444
##    0 | 55666678888999
##    1 | 00011112222233334444
##    1 | 555566666666677777788888889
##    2 | 00011224
##    2 | 9
```

```
#spatial summaries
```

```
plot(s100)
```



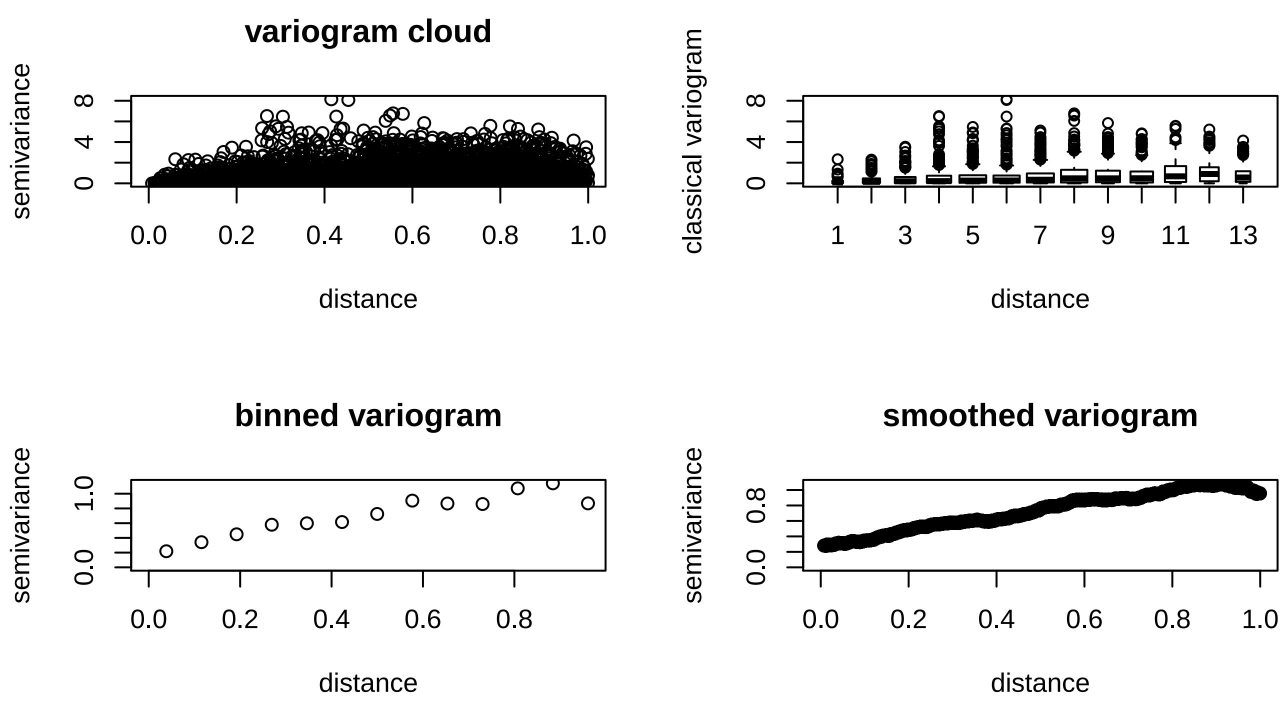

```
#with a 3D plot of the outcome, note that you need to have the package scatterplot3d installed
plot(s100,scatter3d=TRUE)
```




Looking at the empirical variograms

```
#
# empirical variograms variograms:
#

# binned variogram
vario.b <- variog(s100, max.dist=1)

## variog: computing omnidirectional variogram

# variogram cloud
vario.c <- variog(s100, max.dist=1, op="cloud")

## variog: computing omnidirectional variogram
```

```r
#binned variogram and stores the cloud
vario.bc <- variog(s100, max.dist=1, bin.cloud=TRUE)

## variog: computing omnidirectional variogram

# smoothed variogram
vario.s <- variog(s100, max.dist=1, op="sm", band=0.2) # pag 97 do manual geoR

## variog: computing omnidirectional variogram

#
#

# plotting the variograms:
par(mfrow=c(2,2))
plot(vario.c, main="variogram cloud")
plot(vario.bc, bin.cloud=TRUE, main="clouds for binned variogram")
plot(vario.b, main="binned variogram")
plot(vario.s, main="smoothed variogram")
```



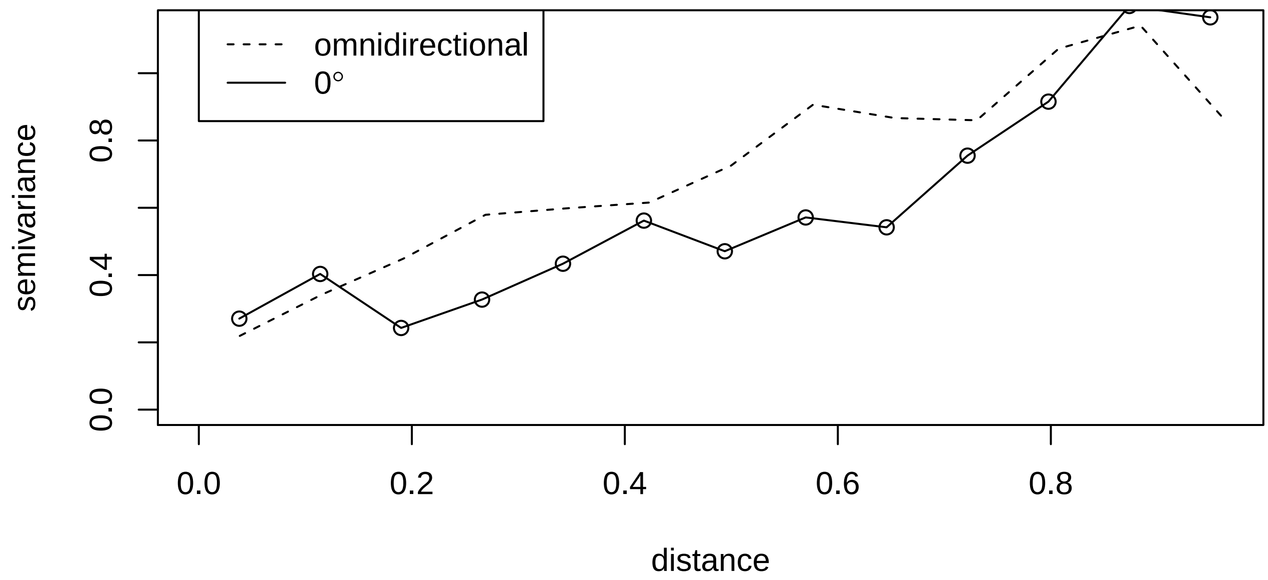

Computing directional variograms

```r
# computing a directional variogram
par(mfrow=c(1,1))
```

```
vario.0 <- variog(s100, max.dist=1, dir=0, tol=pi/8)
```

```
## variog: computing variogram for direction = 0 degrees (0 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
```

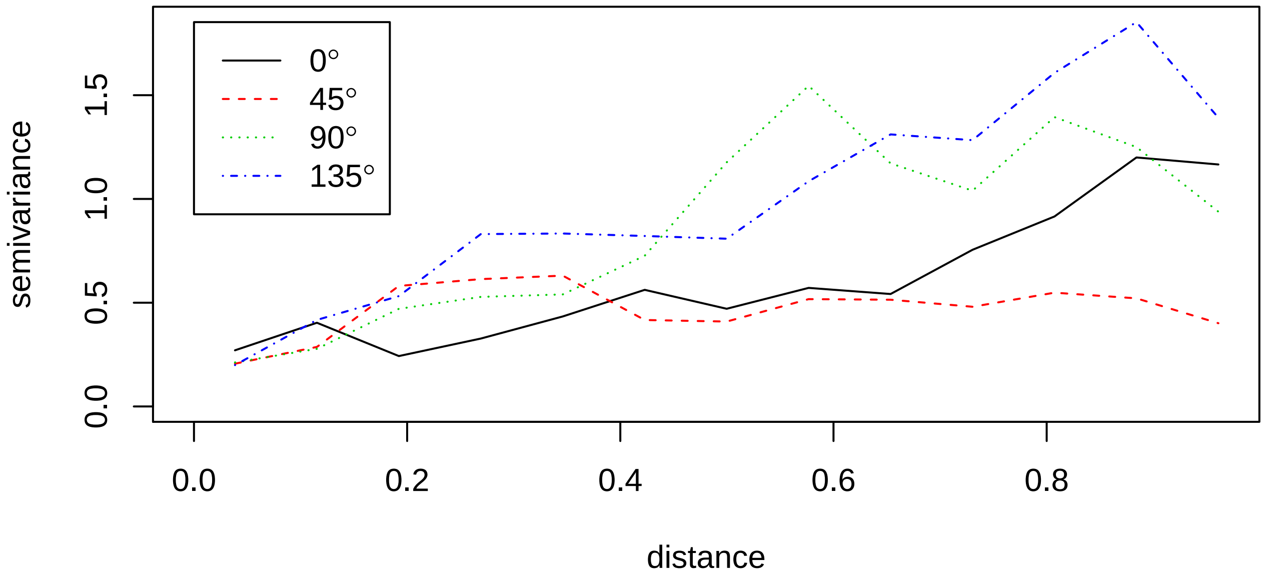
```
plot(vario.b, type="l", lty=2)
lines(vario.0)
legend(0, 1.2, legend=c("omnidirectional", expression(0 * degree)), lty=c(2,1))
```




```
# computing the empirical variogram along directions 0, 45, 90 e 135
var4<-variog4(s100,max.dist=1)
```

```
## variog: computing variogram for direction = 0 degrees (0 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 90 degrees (1.571 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 135 degrees (2.356 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing omnidirectional variogram
```

```
par(mfrow=c(1,1))
plot(var4)
```




## 2 Spatial regression

We start by generating an artificial data on a regular grid with 441 points, variance equals 1 and $\phi = 0.25$.

```
pot<-grf(441,grid="reg",cov.pars=c(1,0.25))

## grf: generating grid  21  *  21  with  441  points
## grf: process with  1  covariance structure(s)
## grf: nugget effect is: tausq= 0
## grf: covariance model 1 is: exponential(sigmasq=1, phi=0.25)
## grf: decomposition algorithm used is:  cholesky
## grf: End of simulation procedure. Number of realizations: 1

image(pot)
```

Now we fit different models assuming different polynomial trends for the mean.

```
ml <- likfit(pot, ini=c(0.1, 1), fix.nug = TRUE)

## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optimize.
## likfit: Use control() to pass additional
```

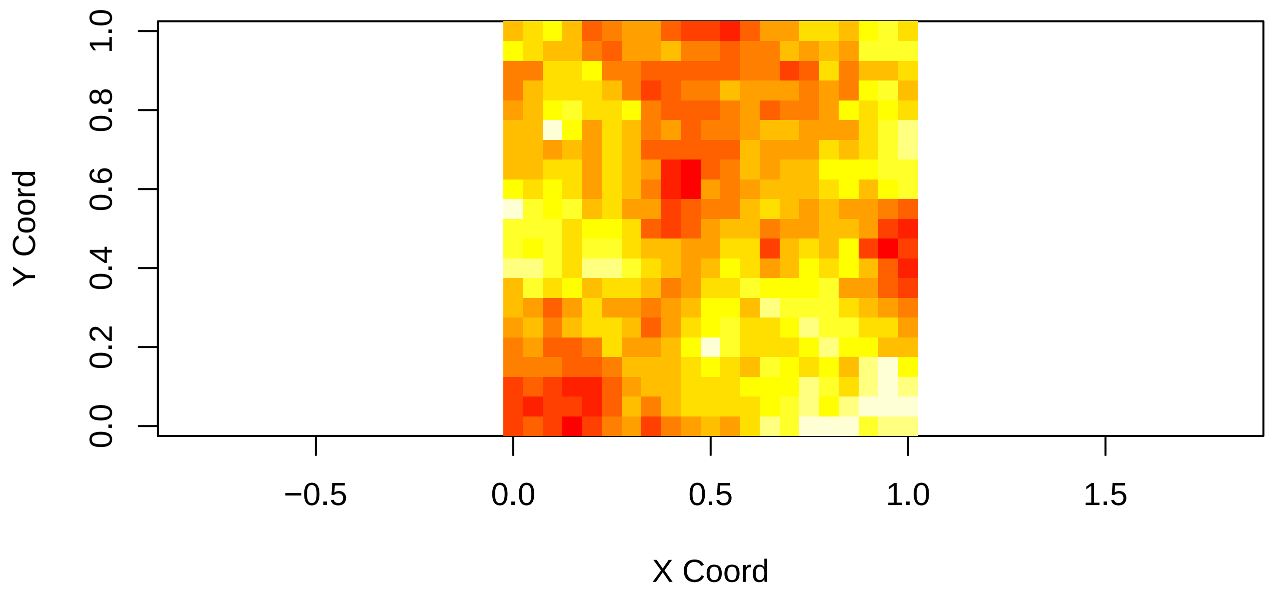



Figure 1: Realization of a GP with exponential correlation function and constant mean.

```
##           arguments for the maximisation function.
##          For further details see documentation for optimize.
## likfit: It is highly advisable to run this function several
##          times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```
summary(ml)
```

```
## Summary of the parameter estimation
## -----------------------------------
## Estimation method: maximum likelihood
##
## Parameters of the mean component (trend):
##   beta
## 0.576
##
## Parameters of the spatial component:
##    correlation function: exponential
```

```
##       (estimated) variance parameter sigmasq (partial sill) =  0.7691
##       (estimated) cor. fct. parameter phi (range parameter)  =  0.2164
##    anisotropy parameters:
##       (fixed) anisotropy angle = 0  ( 0 degrees )
##       (fixed) anisotropy ratio = 1
##
## Parameter of the error component:
##       (fixed) nugget = 0
##
## Transformation parameter:
##       (fixed) Box-Cox parameter = 1 (no transformation)
##
## Practical Range with cor=0.05 for asymptotic range: 0.648429
##
## Maximised Likelihood:
##    log.L n.params      AIC      BIC
## "-266.1"      "3"  "538.3"  "550.6"
##
## non spatial model:
##    log.L n.params      AIC      BIC
## "-532.2"      "2"   "1068"   "1077"
##
## Call:
## likfit(geodata = pot, ini.cov.pars = c(0.1, 1), fix.nugget = TRUE)
```

```r
ml1<-likfit(pot, trend=trend.spatial("1st",pot),ini=c(0.5, 0.5), fix.nug = TRUE)
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optimize.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optimize.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```r
summary(ml1)
```

```
## Summary of the parameter estimation
```

```
## ---------------------------------------
## Estimation method: maximum likelihood
##
## Parameters of the mean component (trend):
##   beta0   beta1   beta2
##  0.3260  0.6913 -0.1837
##
## Parameters of the spatial component:
##    correlation function: exponential
##       (estimated) variance parameter sigmasq (partial sill) =  0.7101
##       (estimated) cor. fct. parameter phi (range parameter)  =  0.199
##    anisotropy parameters:
##       (fixed) anisotropy angle = 0  ( 0 degrees )
##       (fixed) anisotropy ratio = 1
##
## Parameter of the error component:
##       (fixed) nugget = 0
##
## Transformation parameter:
##       (fixed) Box-Cox parameter = 1 (no transformation)
##
## Practical Range with cor=0.05 for asymptotic range: 0.5961362
##
## Maximised Likelihood:
##    log.L n.params      AIC      BIC
## "-265.5"      "5"  "541.1"  "561.5"
##
## non spatial model:
##    log.L n.params      AIC      BIC
## "-515.5"      "4"   "1039"   "1055"
##
## Call:
## likfit(geodata = pot, trend = trend.spatial("1st", pot), ini.cov.pars = c(0.5,
##     0.5), fix.nugget = TRUE)
```

```
# ajustando modelo com media polinomio 2a ordem e funcao de correlacao exponencial
ml2<-likfit(pot, trend=trend.spatial("2nd",pot),ini=c(0.5, 0.5), fix.nug = TRUE)
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optimize.
```

```
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optimize.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```
summary(ml2)
```

```
## Summary of the parameter estimation
## -----------------------------------
## Estimation method: maximum likelihood
##
## Parameters of the mean component (trend):
##   beta0   beta1   beta2   beta3   beta4   beta5
## -0.1752  1.0631  1.5072  0.6578 -0.6678 -2.0804
##
## Parameters of the spatial component:
##    correlation function: exponential
##       (estimated) variance parameter sigmasq (partial sill) =  0.6214
##       (estimated) cor. fct. parameter phi (range parameter)  =  0.1726
##    anisotropy parameters:
##       (fixed) anisotropy angle = 0  ( 0 degrees )
##       (fixed) anisotropy ratio = 1
##
## Parameter of the error component:
##       (fixed) nugget = 0
##
## Transformation parameter:
##       (fixed) Box-Cox parameter = 1 (no transformation)
##
## Practical Range with cor=0.05 for asymptotic range: 0.5172129
##
## Maximised Likelihood:
##    log.L n.params      AIC      BIC
## "-264.3"      "8"  "544.7"  "577.4"
##
## non spatial model:
```

```
##    log.L n.params      AIC      BIC
## "-483.1"      "7"  "980.2"   "1009"
##
## Call:
## likfit(geodata = pot, trend = trend.spatial("2nd", pot), ini.cov.pars = c(0.5,
##     0.5), fix.nugget = TRUE)
```

```r
par(mfrow=c(2,2))
plot(variog(pot),max.dist=1.1,bty="n")

## variog: computing omnidirectional variogram

lines(ml)
plot(variog(pot),max.dist=1.1,bty="n")

## variog: computing omnidirectional variogram

lines(ml1)
plot(variog(pot),max.dist=1.1,bty="n")

## variog: computing omnidirectional variogram

lines(ml2)
plot(variog(pot),max.dist=1.1,bty="n")

## variog: computing omnidirectional variogram

lines(ml,lty=1)
lines(ml1,lty=2)
lines(ml2,lty=3)
```

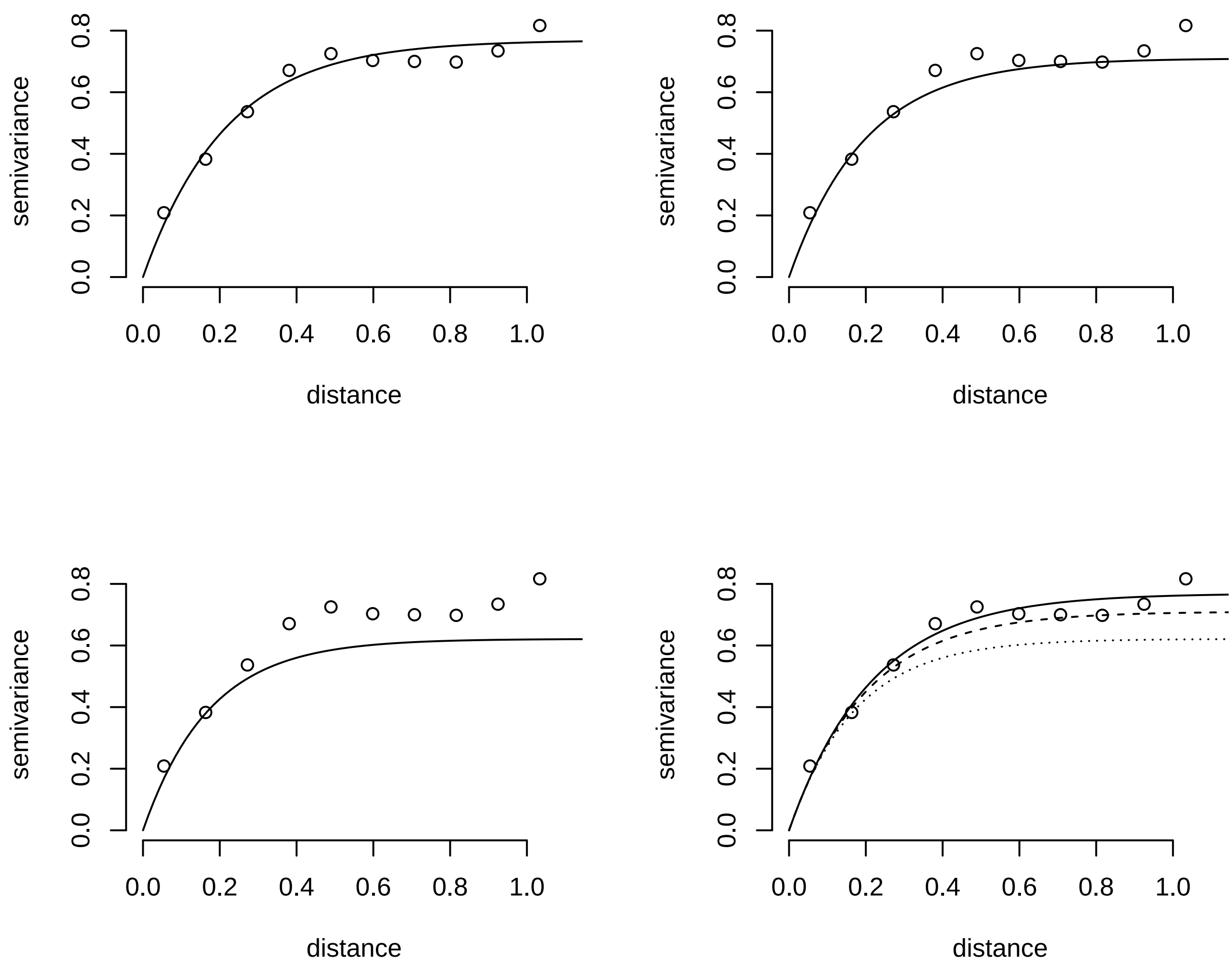



Figure 2: Empirical and fitted variograms under constant mean, first order trend and second order trend.

```r
citation(package="geoR")
```

```
##
## To cite package 'geoR' in publications use:
##
##   Paulo J. Ribeiro Jr and Peter J. Diggle (2016). geoR: Analysis
##   of Geostatistical Data. R package version 1.7-5.2.
##   https://CRAN.R-project.org/package=geoR
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {geoR: Analysis of Geostatistical Data},
##     author = {Paulo J. {Ribeiro Jr} and Peter J. Diggle},
##     year = {2016},
##     note = {R package version 1.7-5.2},
##     url = {https://CRAN.R-project.org/package=geoR},
##   }
##
## ATTENTION: This citation information has been auto-generated from
## the package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```