Stockton dataset

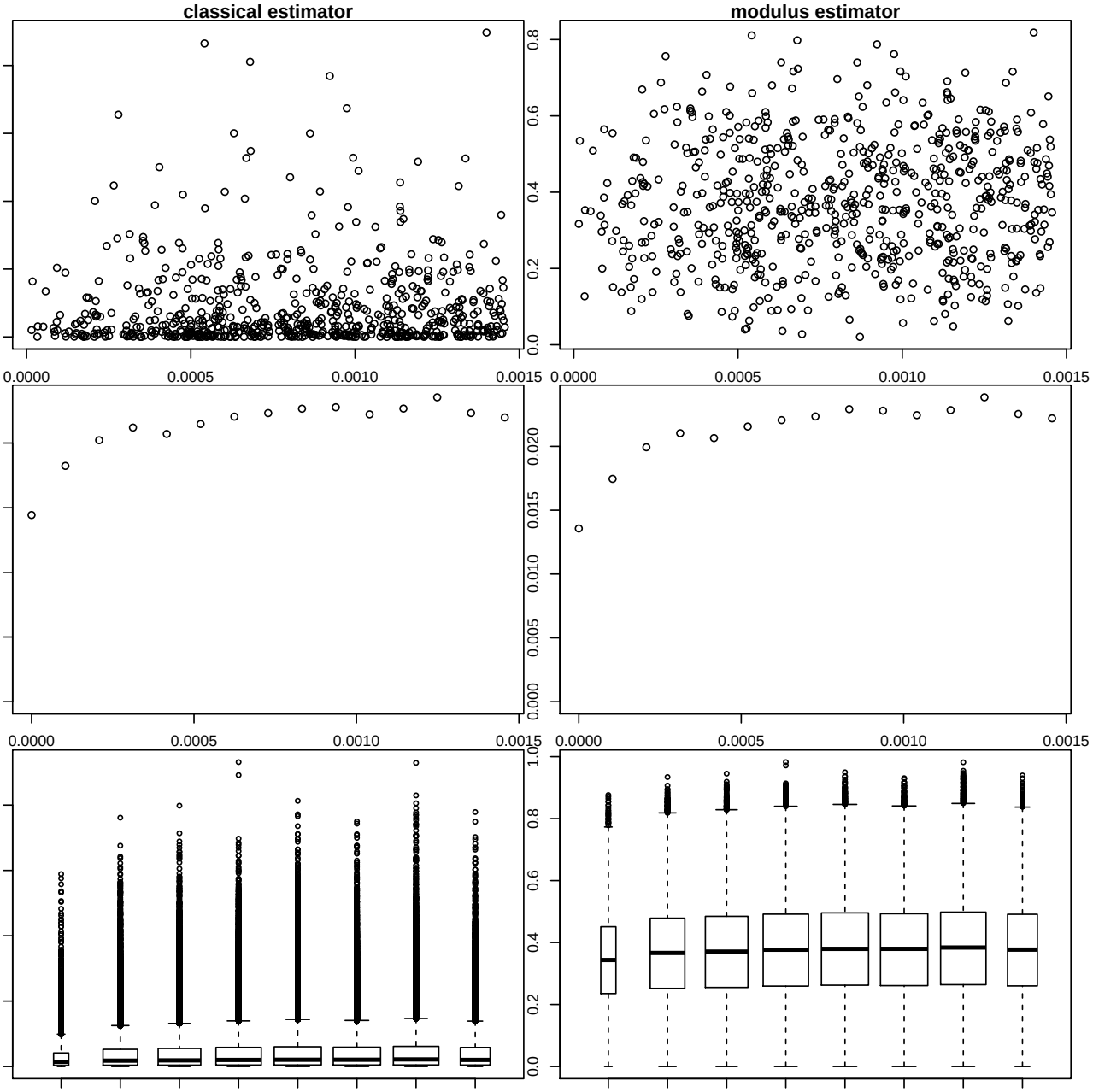
Contents

***All below data are de-trended.**

Created on January 21, 2014, with `R version 2.14.1 (2011-12-22)` on an `x86_64-pc-linux-gnu` machine.

1 variograms

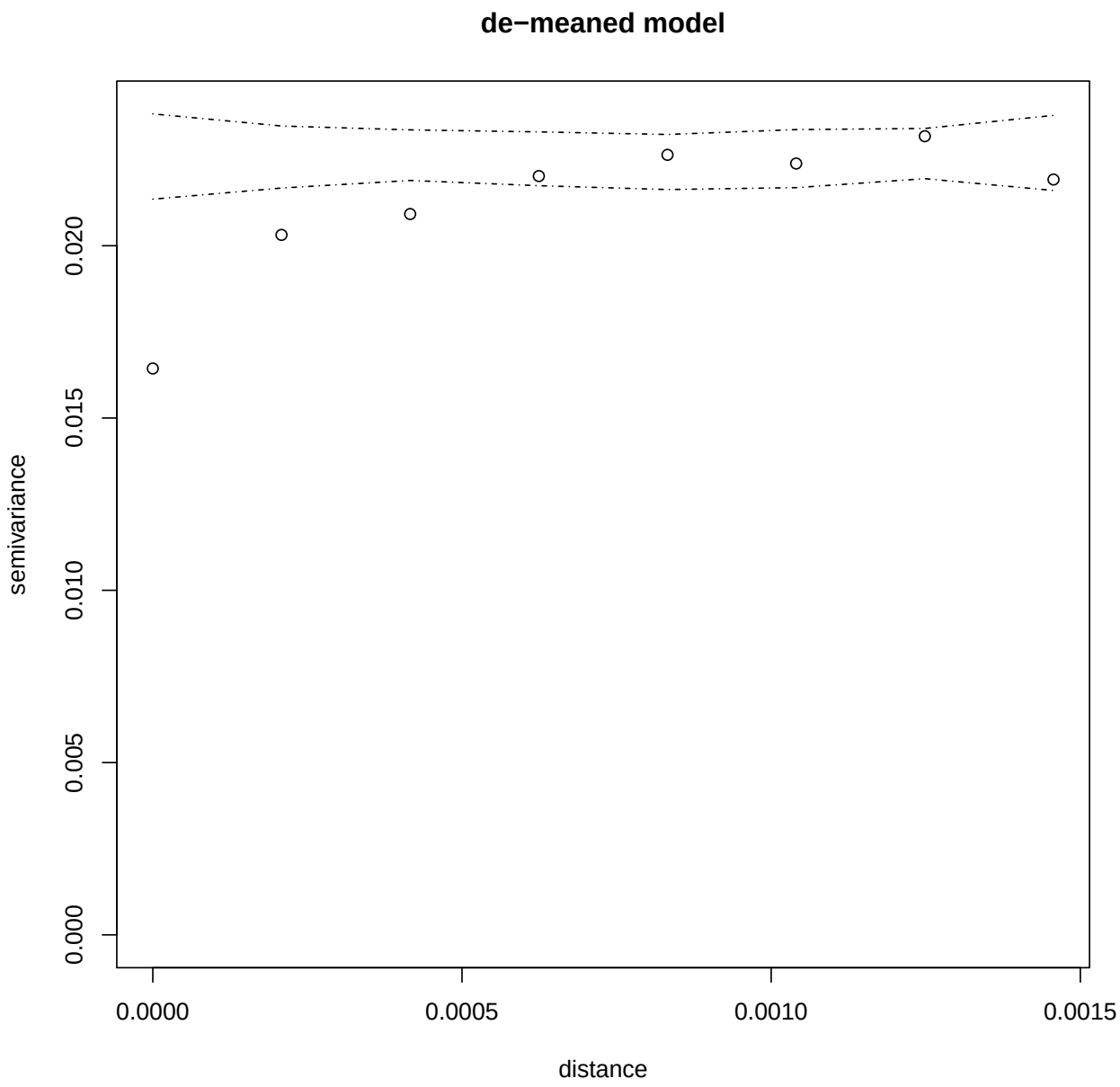
1.1 variogram figures



1.2 variogram envelopes

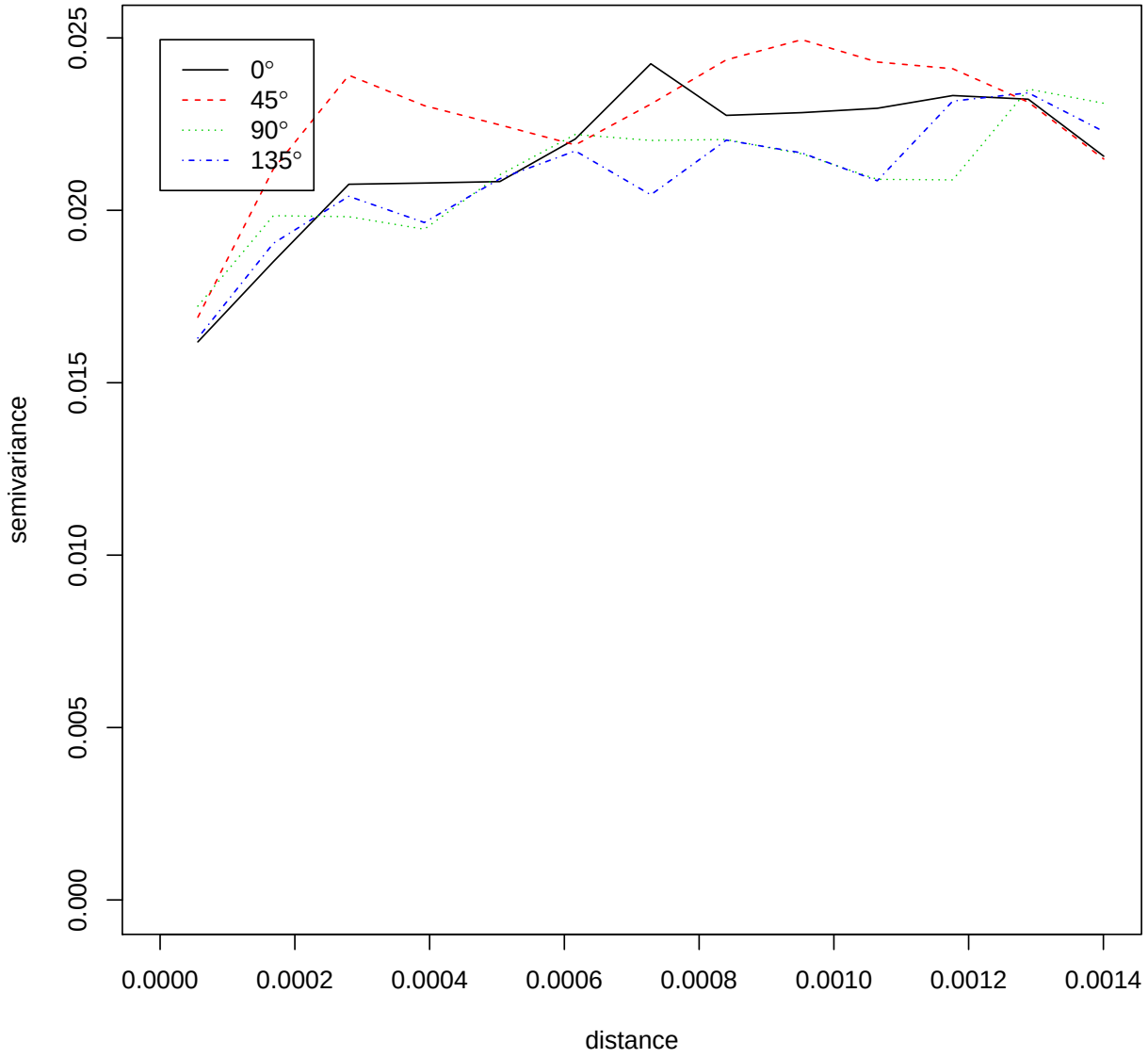
"to test the constant mean model"

```
> set.seed(1)
> env1R <- variog.mc.env(s1R.geo, ob=bin1R)

variog.env: generating 99 simulations by permutating data values
variog.env: computing the empirical variogram for the 99 simulations
variog.env: computing the envelops

> plot(bin1R, env=env1R, main="de-meaned model")
> rm(bin1R,bin2R)
```



1.3 directional variogram

```
> vario4R <- variog4(s1R.geo, max.dist = maxD)
> plot(vario4R)
```



2 estimating parameters

2.1 estimating parameters with variofit()

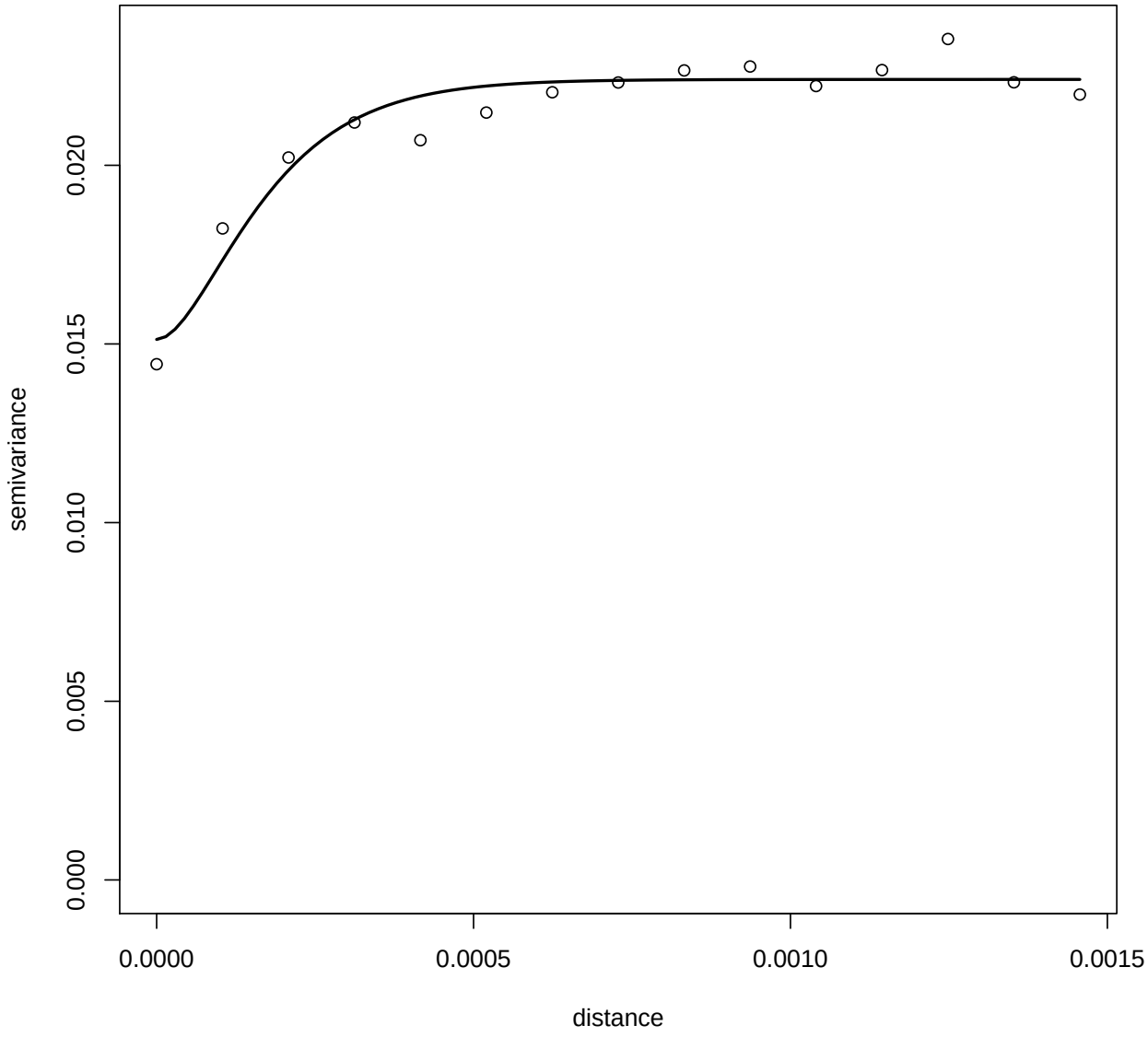
```
> pSill <- .00665
> phi   <- .0001
> bin   <- variog(s1R.geo, uvec=seq(0,maxD,l=10),
+                 max.dist=maxD)

variog: computing omnidirectional variogram

> variogk15 <- variofit(bin, cov.model='mat',
+                       ini.cov.pars=c(pSill, phi),
+                       kappa=1.5,
+                       nugget=.014,
+                       fix.nugget=FALSE)

variofit: covariance model used is matern
variofit: weights used: npairs
variofit: minimisation function used: optim

> summary(variogk15)

$pmethod
[1] "WLS (weighted least squares)"

$cov.model
[1] "matern"

$spatial.component
     sigmasq          phi
7.277033e-03 9.361362e-05

$spatial.component.extra
kappa
  1.5

$nugget.component
     tausq
0.01512544

$fix.nugget
[1] FALSE

$fix.kappa
[1] TRUE

$practicalRange
[1] 0.0004660384

$sum.of.squares
    value
0.1846355

$estimated.pars
       tausq      sigmasq          phi
1.512544e-02 7.277033e-03 9.361362e-05

$weights
[1] "npairs"

$call
```

```
variofit(vario = bin, ini.cov.pars = c(pSill, phi), cov.model = "mat",
    fix.nugget = FALSE, nugget = 0.014, kappa = 1.5)

attr(,"class")
[1] "summary.variomodel"

> rm(bin,phi,pSill)
```

We see here that for $\kappa = 1.5$, `variofit()` suggests the Matern correlation function

$$\rho(u) = \frac{1}{\sqrt{2}\Gamma(\frac{3}{2})} \cdot \left(\frac{u}{9.4 \times 10^{-5}}\right)^{\frac{3}{2}} K_\kappa \left(\frac{u}{9.4 \times 10^{-5}}\right)$$

with $\tau^2 = 0.01512$. We can quickly plot that against the empirical variogram:



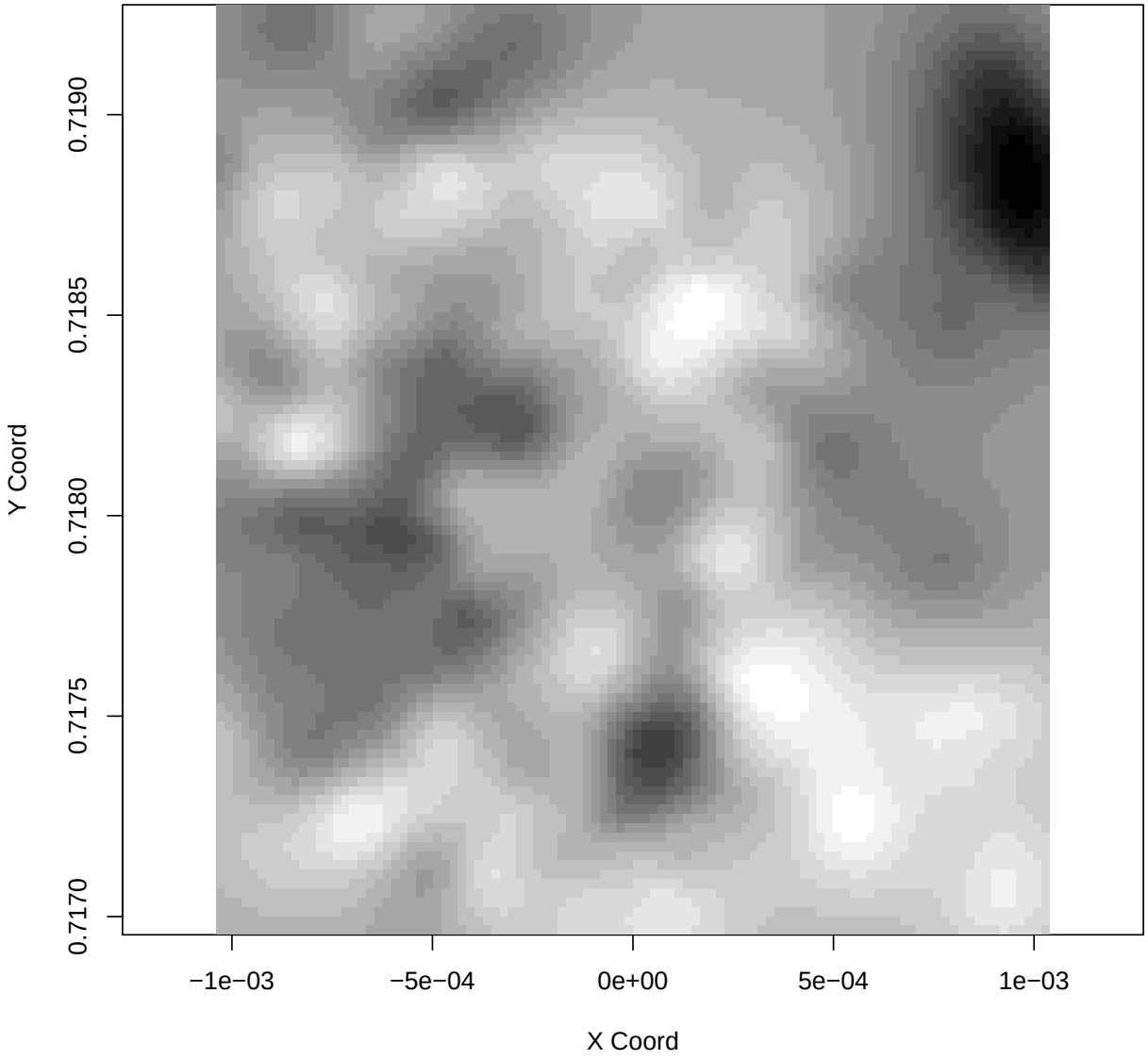
Looks like a pretty good fit.

2.2 likfit()

```
> mlfit <- likfit(s1R.geo, cov.model='mat',
+                 ini=c(7.277033e-03, 9.361362e-05),
+                 kap=1.5, nug=1.512544e-02)

---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
         arguments for the maximisation function.
        For further details see documentation for optim.
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.

> mlfit

likfit: estimated model parameters:
    beta    tausq  sigmasq      phi
"0.0060" "0.0148" "0.0071" "0.0001"
Practical Range with cor=0.05 for asymptotic range: 0.0004660384

likfit: maximised log-likelihood = 774.7

> #summary(mlfit)
```

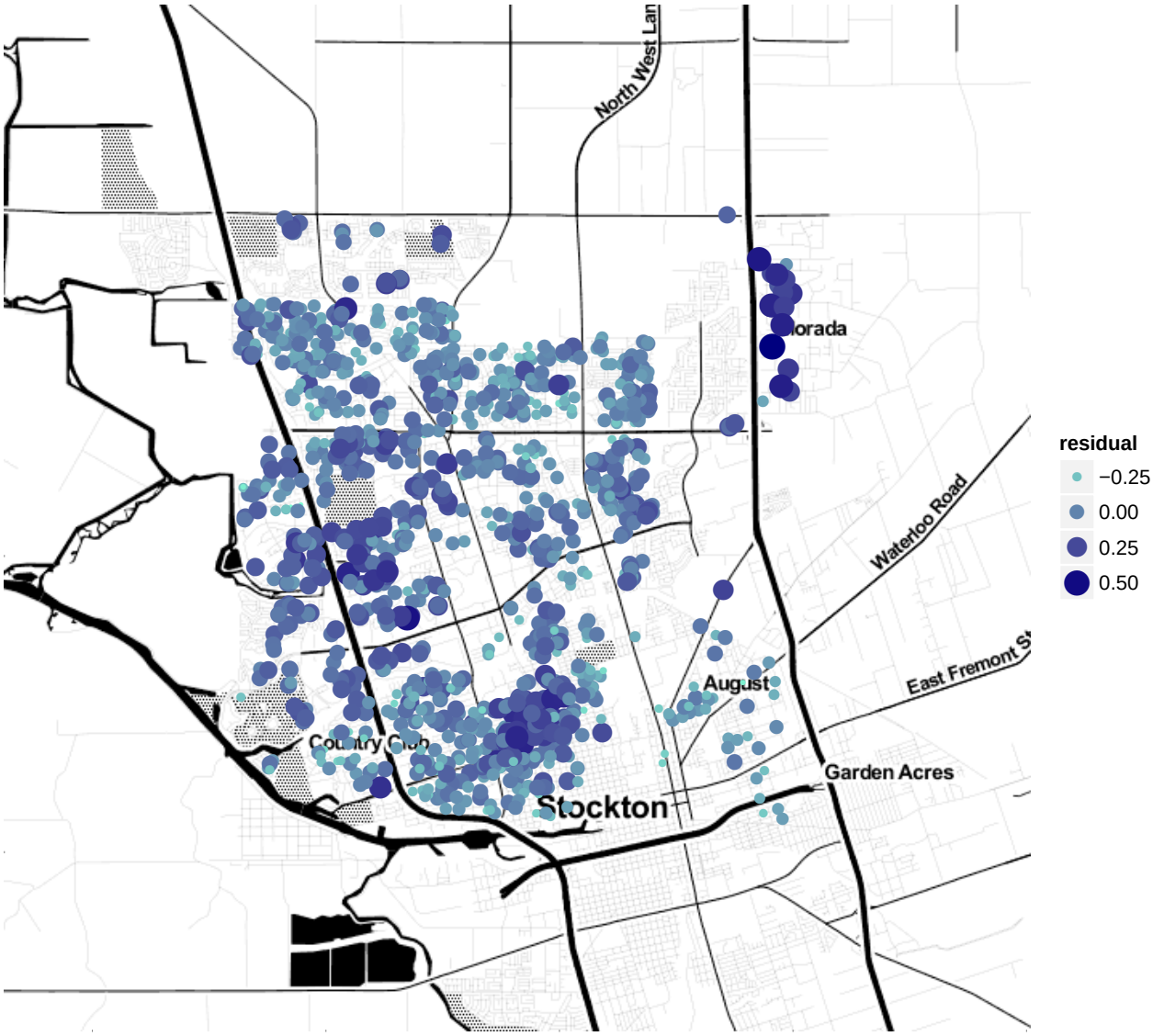
2.3 simple kriging

```
> grid   <- expand.grid(seq(min(s1R$x),max(s1R$x),l=100),
+                       seq(min(s1R$y),max(s1R$y),l=100))
> kc     <- krige.control(obj.model = mlfit)
> set.seed(1)
> pred   <- krige.conv(s1R.geo, loc=grid, krige=kc)

krige.conv: model with constant mean
krige.conv: Kriging performed using global neighbourhood

> image1 <-image(pred, loc = grid, col = gray(seq(1, 0, l=21)))
```



2.4 comparing to our original figure

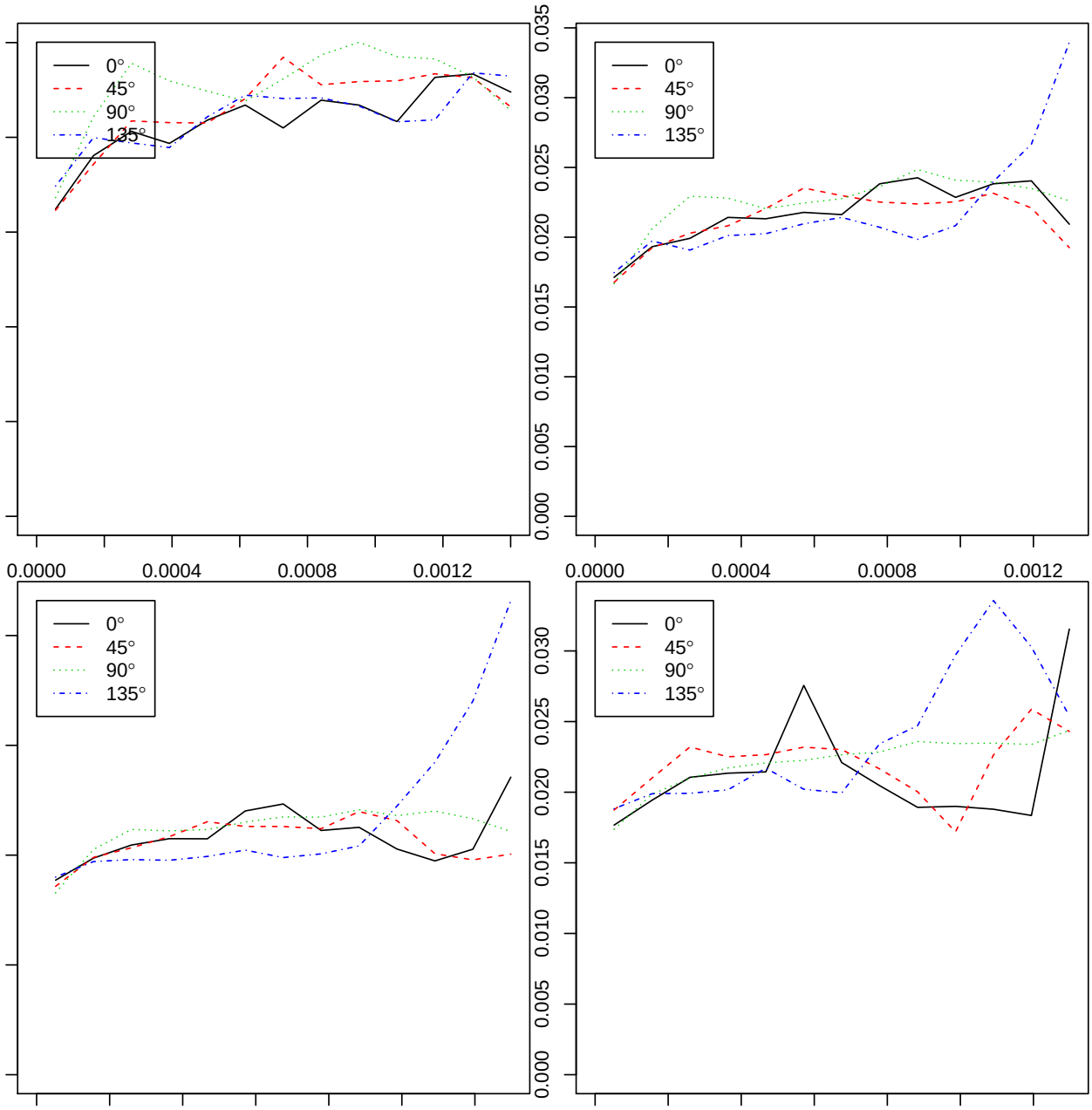
```
> require(ggmap)
> paste0 <- function(..., collapse = NULL) paste(..., sep = "", collapse = collapse)
> cLat=(37.95564 + 38.05752)/2;  cLon=(-121.3681 -121.2502)/2
> stocktonMap_toner <- qmap(c(cLon,cLat), zoom=12, source = "stamen",
+                           maptype = "toner")
> stocktonMap_toner +
+   geom_point(aes(x=long, y=lat, colour=residual, size=residual), data=s) +
+   scale_colour_gradient(low = "aquamarine1", high = "navyblue") +
+   guides(colour = guide_legend())
> rm(s)
```



3 assuming anisotropy (azimuthal angle = 135 deg)

3.1 anisotropic directional variogram

`coords.aniso()` takes two arguments: anisotropy angle "psiA" (which here is 135 degrees = $\frac{3\pi}{4}$ radians), and anisotropy ratio "psiR" which I've varied in the below: 1 (no change), 1.5, 2, 5:



3.2 anisotropic log-likelihoods (with loglik.GRF())

Looking at the log likelihoods for each of the above:

```
> # log likelihood for basic model
> loglik.GRF(s1R.geo, cov.model='mat',
+            cov.pars=c(7.277033e-03, 9.361362e-05),
+            kap=1.5, nug=1.512544e-02,
+            psiA=0, psiR=1)

[1] 774.5323

> # log likelihood for model, psiA=(3*pi)/4, psiR=1.5
> loglik.GRF(s1R.geo, cov.model='mat',
```

```
+            cov.pars=c(7.277033e-03, 9.361362e-05),
+            kap=1.5, nug=1.512544e-02,
+            psiA=(3*pi)/4, psiR=1.5)

[1] 766.9429

> # log likelihood for model, psiA=(3*pi)/4, psiR=2
> loglik.GRF(s1R.geo, cov.model='mat',
+            cov.pars=c(7.277033e-03, 9.361362e-05),
+            kap=1.5, nug=1.512544e-02,
+            psiA=(3*pi)/4, psiR=2)

[1] 758.4905

> # log likelihood for model, psiA=(3*pi)/4, psiR=5
> loglik.GRF(s1R.geo, cov.model='mat',
+            cov.pars=c(7.277033e-03, 9.361362e-05),
+            kap=1.5, nug=1.512544e-02,
+            psiA=(3*pi)/4, psiR=5)

[1] 724.8255
```

All these numbers seem very similar.
Given the similarity, I wanted to check with you both before chopping the points up into quadrants and re-running. (I also ran out of time.) :)