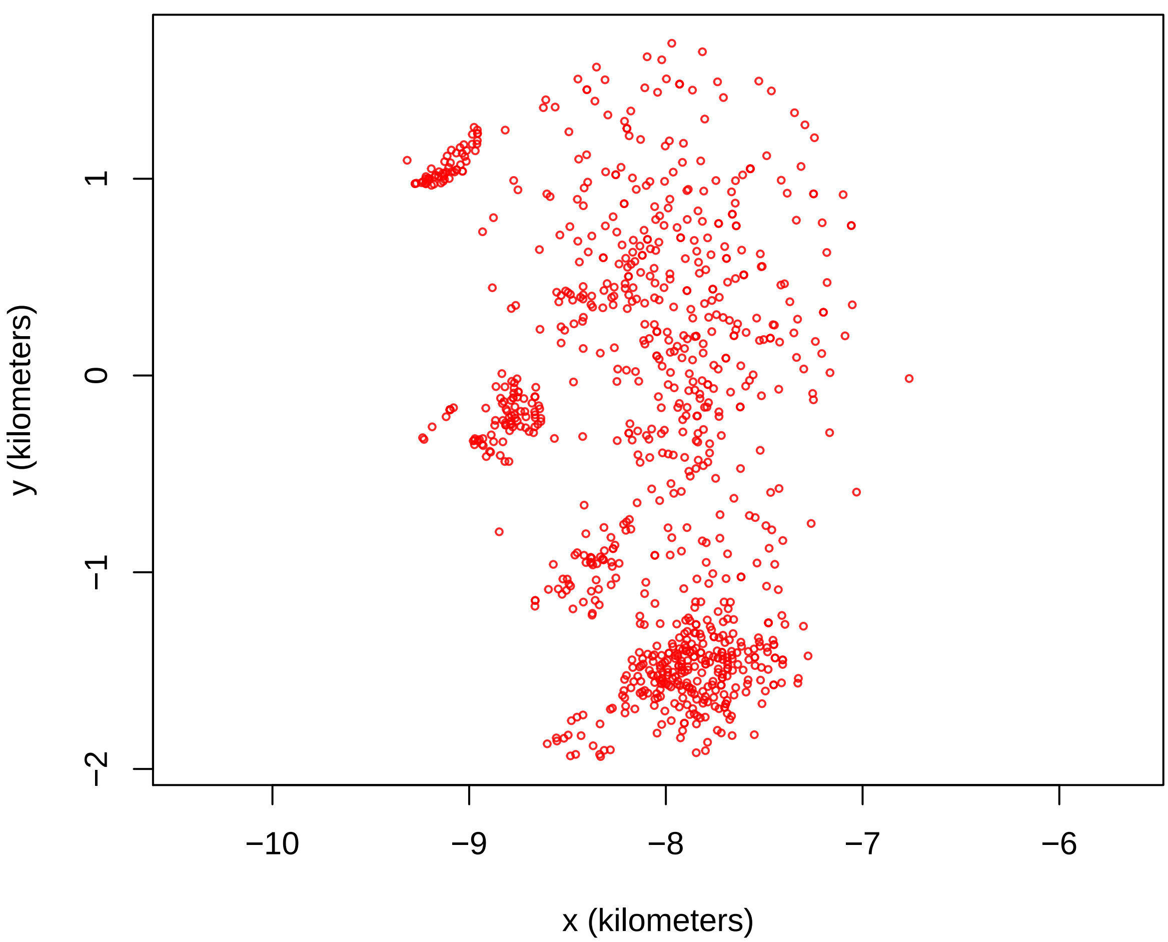
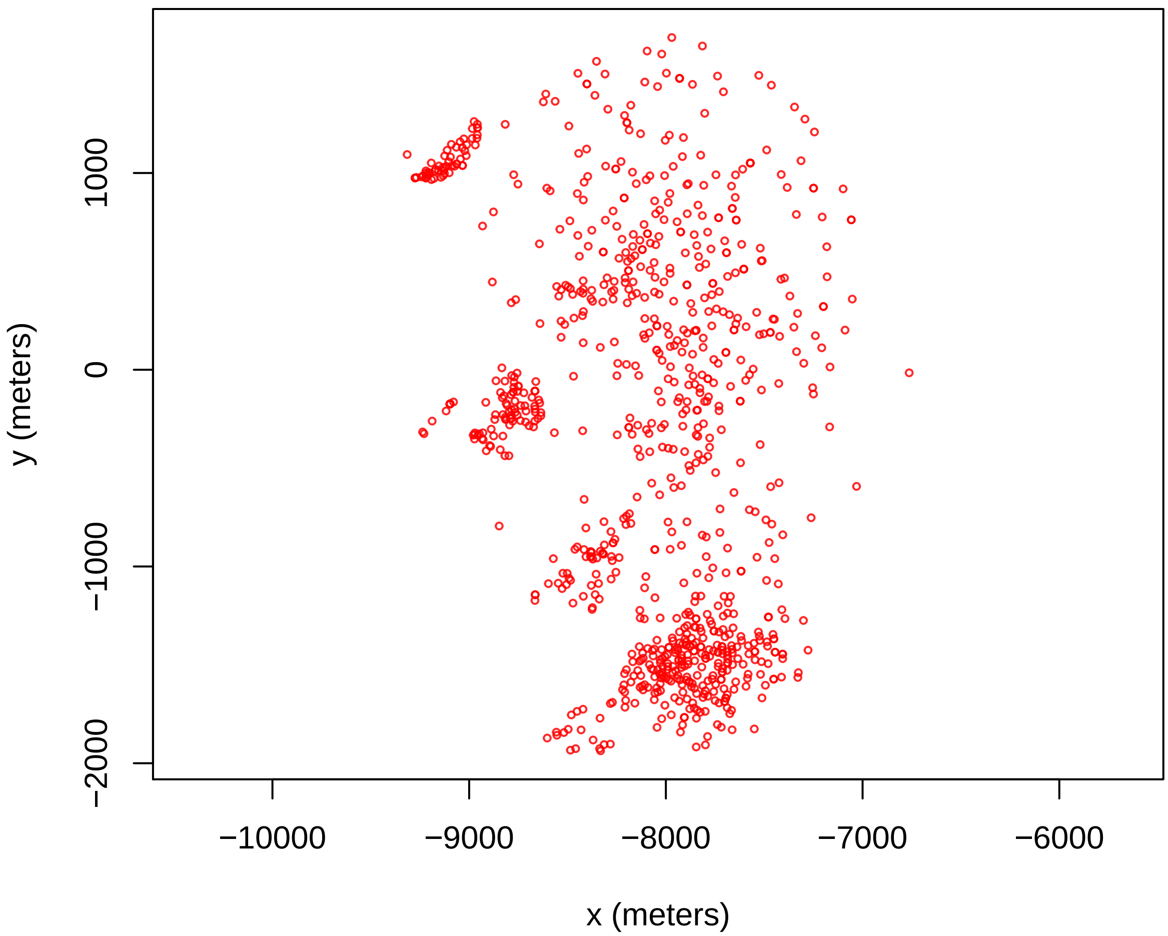
5 main steps to the analysis:

1. import data and convert to `telemetry` format, then plot by position and time

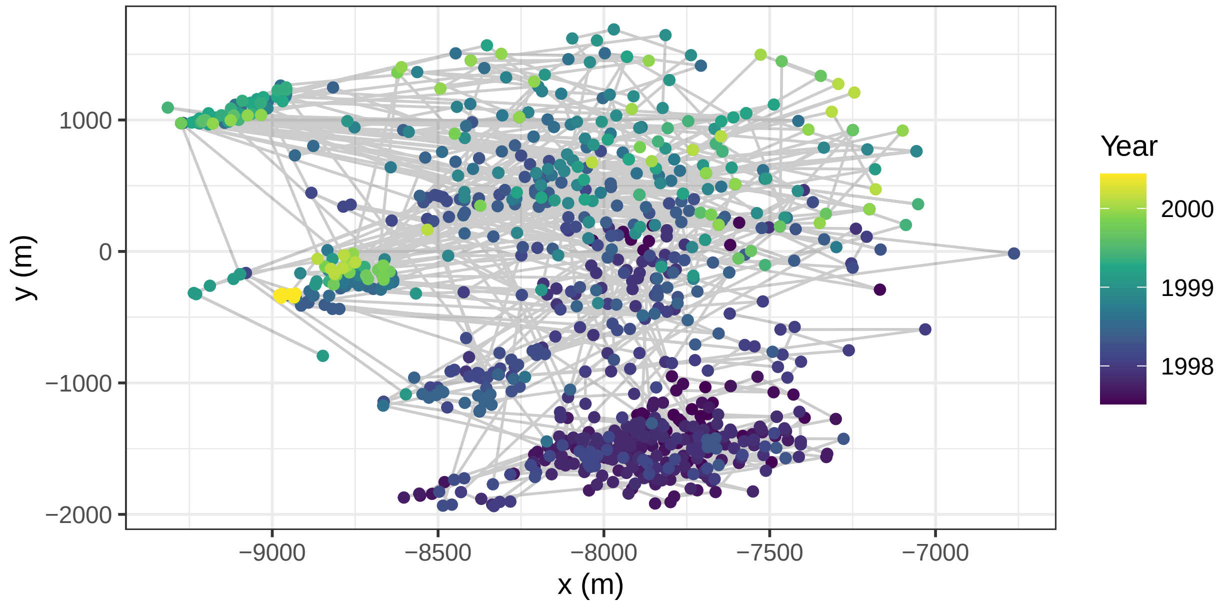
```
library('ctmm') # using the github version (0.6.1)
library('dplyr')
library('lubridate')
library('ggplot2')
theme_set(theme_bw())

tapirs <-
  read.csv('../data/1_ATLANTICFOREST_11.csv') %>%
  as.telemetry(timeformat = '%Y-%m-%d %H:%M:%S')

joana <- tapirs[['AF_01_JOANA']] # take the first tapir

plot(joana) # default units; no information on errors, so assume DOP = 1
```



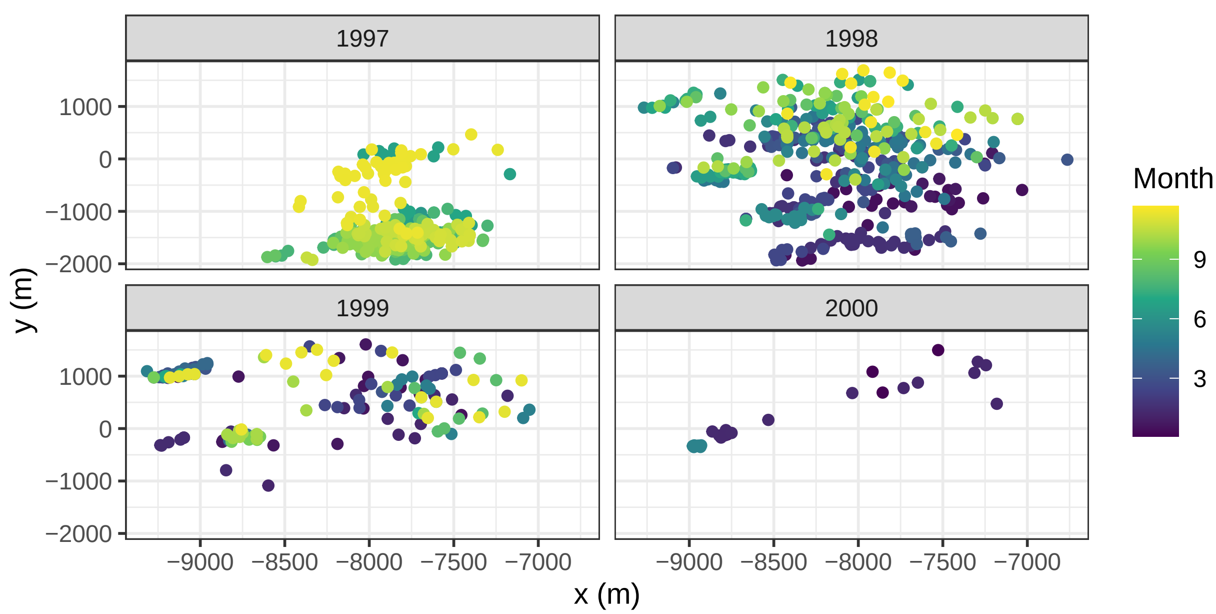
```r
plot(joana, units = FALSE) # force SI units
```




```r
# non-stationary movement (plot is not north-oriented)
ggplot(joana, aes(x, y)) +
  geom_path(alpha = 0.2) +
  geom_point(aes(color = decimal_date(timestamp))) +
  scale_color_viridis_c('Year', breaks = c(1998:2000)) +
  labs(x = 'x (m)', y = 'y (m)')
```

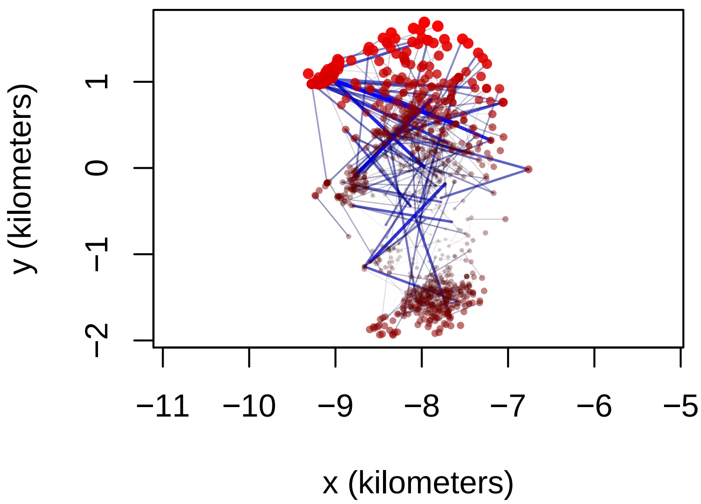



```
# facet plot to see seasonal detail
ggplot(joana, aes(x, y)) +
  facet_wrap(~ year(timestamp)) +
  geom_point(aes(color = (decimal_date(timestamp) %% 1) * 12)) +
  scale_color_viridis_c('Month', breaks = c(3, 6, 9)) +
  labs(x = 'x (m)', y = 'y (m)')
```

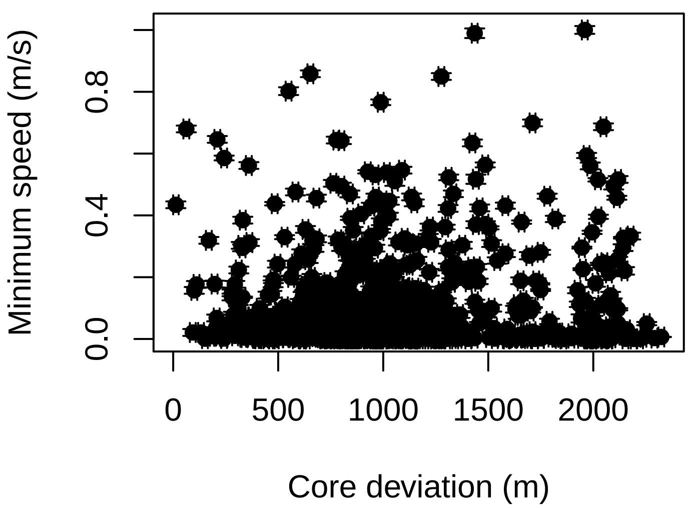


2. check for outliers

```
diagn <- outlie(joana) # some displacement outliers, one velocity outlier
```

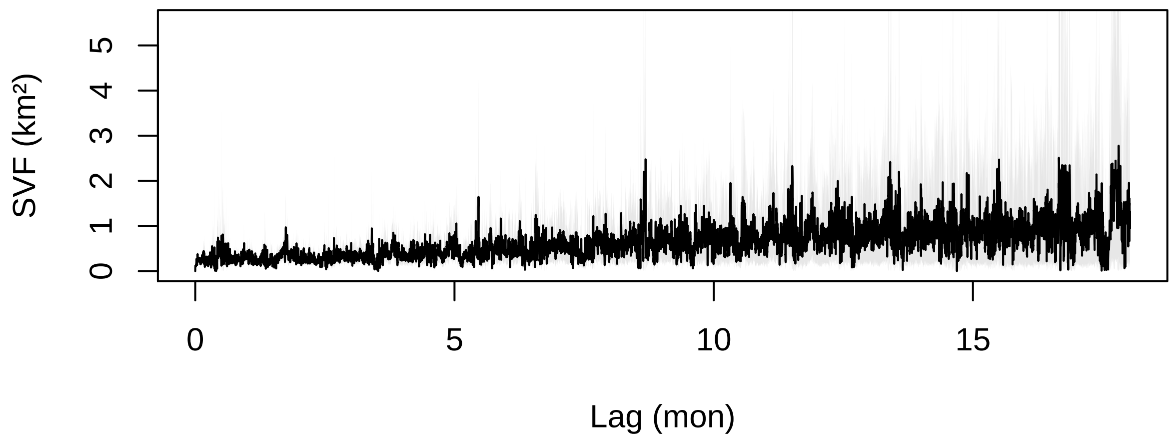



```
plot(diagn, units = FALSE) # very narrow CIs
```

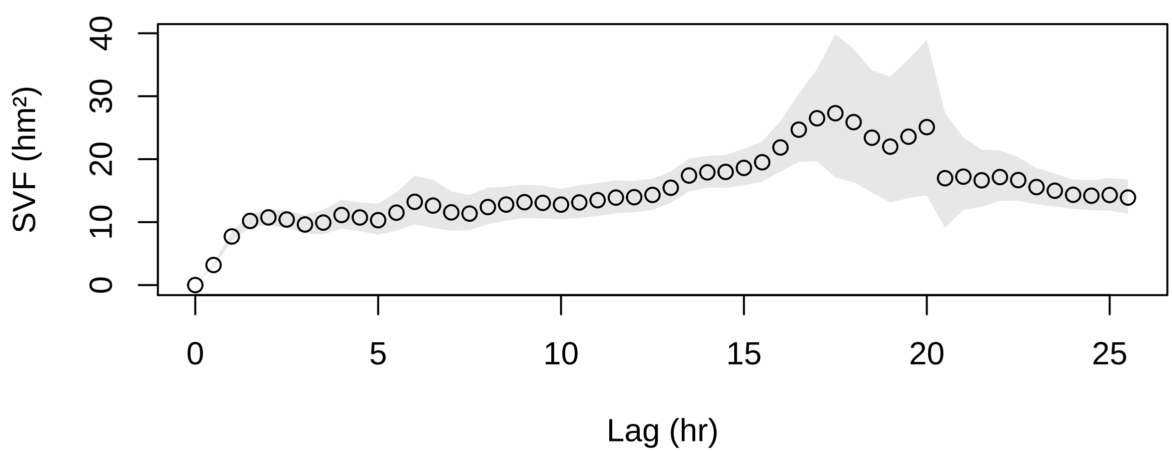


3. Variograms

```
svf <- variogram(joana) # estimate semi-variance function
head(warnings(), n = 2) # some warnings occur occasionally
# NULL
plot(svf) # plots 50% max lag by default, > 50% is "usually garbage"
```




```
plot(svf, fraction = 1e-3) # zoom in on the shortest lags
```



4. fit and select movement model

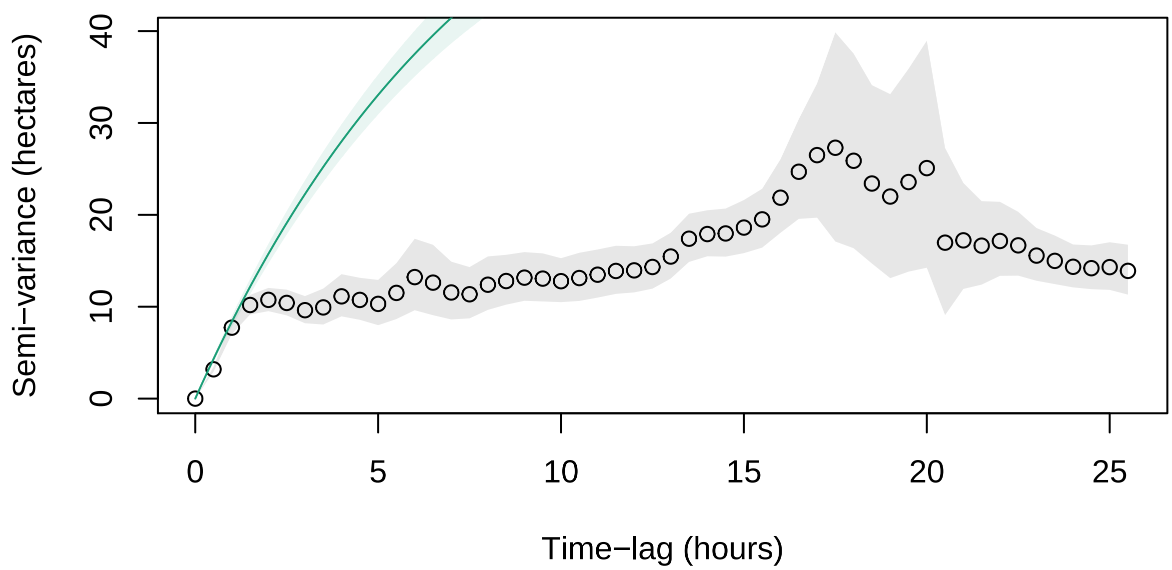
```
variogram.fit(svf, name = 'theta0', fraction = 0.0003)
fitted.mods <- ctmm.select(joana, CTMM = theta0, verbose = TRUE, cores = 0)
```

Negligible difference between OU and OUF models (both anisotropic and isotropic), but choose anisotropic since the tapir's movement is not directionally uniform.

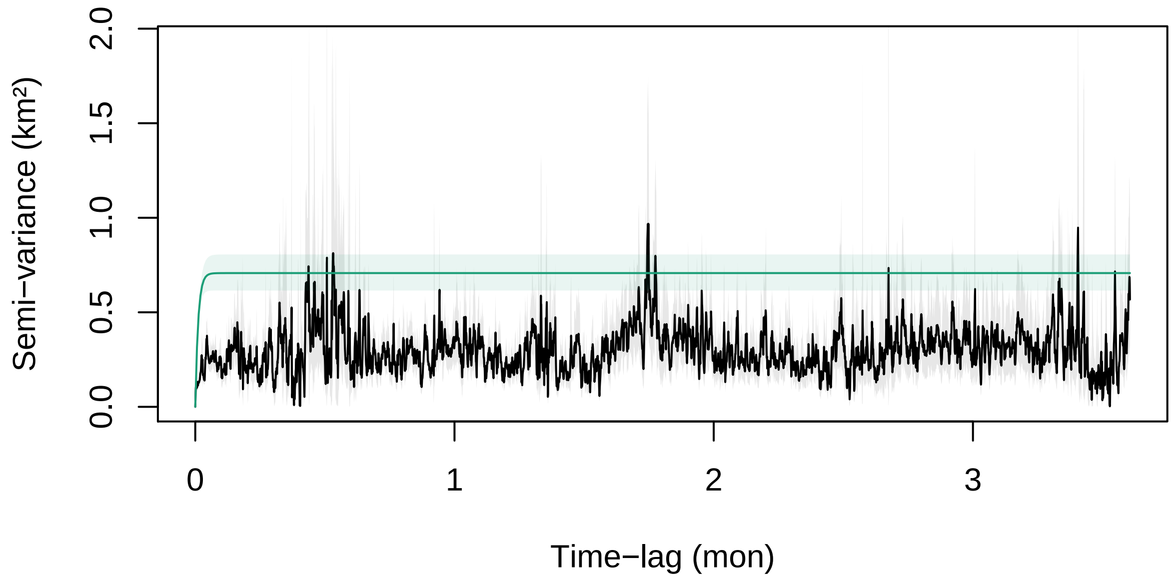
```
summary(fitted.mods)
#                         <U+0394>AICc        <U+0394>RMSPE (m) DOF[area]
# OU anisotropic      0.000000         153.35132  216.7769
# OU isotropic        1.480760         123.15327  248.7336
# OUF anisotropic     2.015385         149.91268  220.0352
# OUF isotropic       3.488707         120.41786  252.1144
# OUf anisotropic   465.039453           0.00000  408.7265
# IID anisotropic  1955.215875          70.34647  955.3603
OUa <- fitted.mods$`OU anisotropic` # Ornstein-Uhlenbeck
```

5. calculate distance and speed estimates

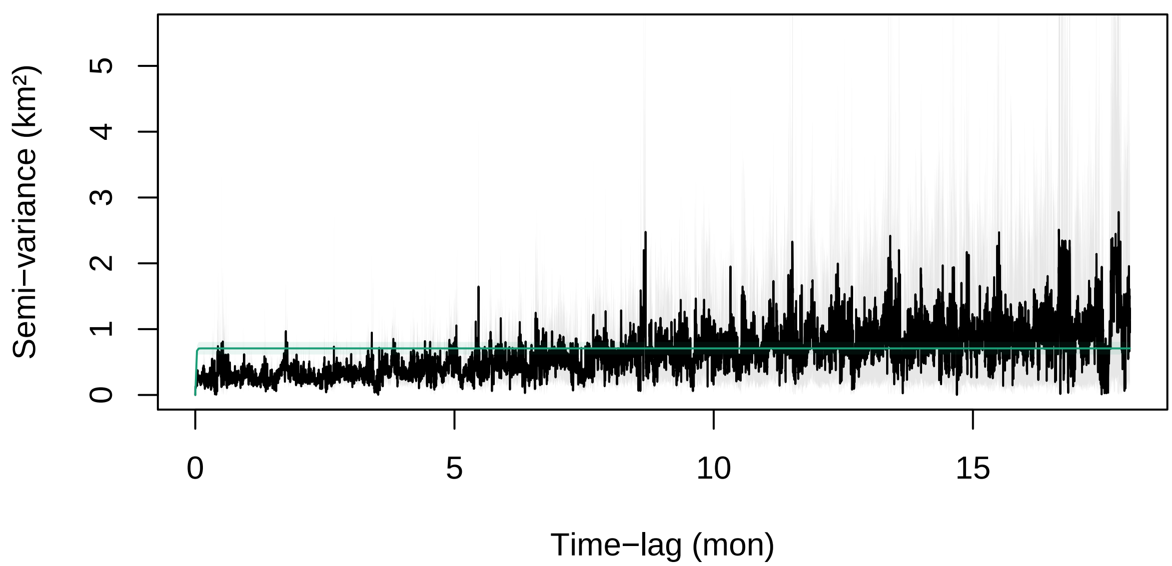
```
# plot semivariance function against empirical variogram
plot(svf, CTMM = OUa, col.CTMM = '#1b9e77', fraction = 0.001)
```



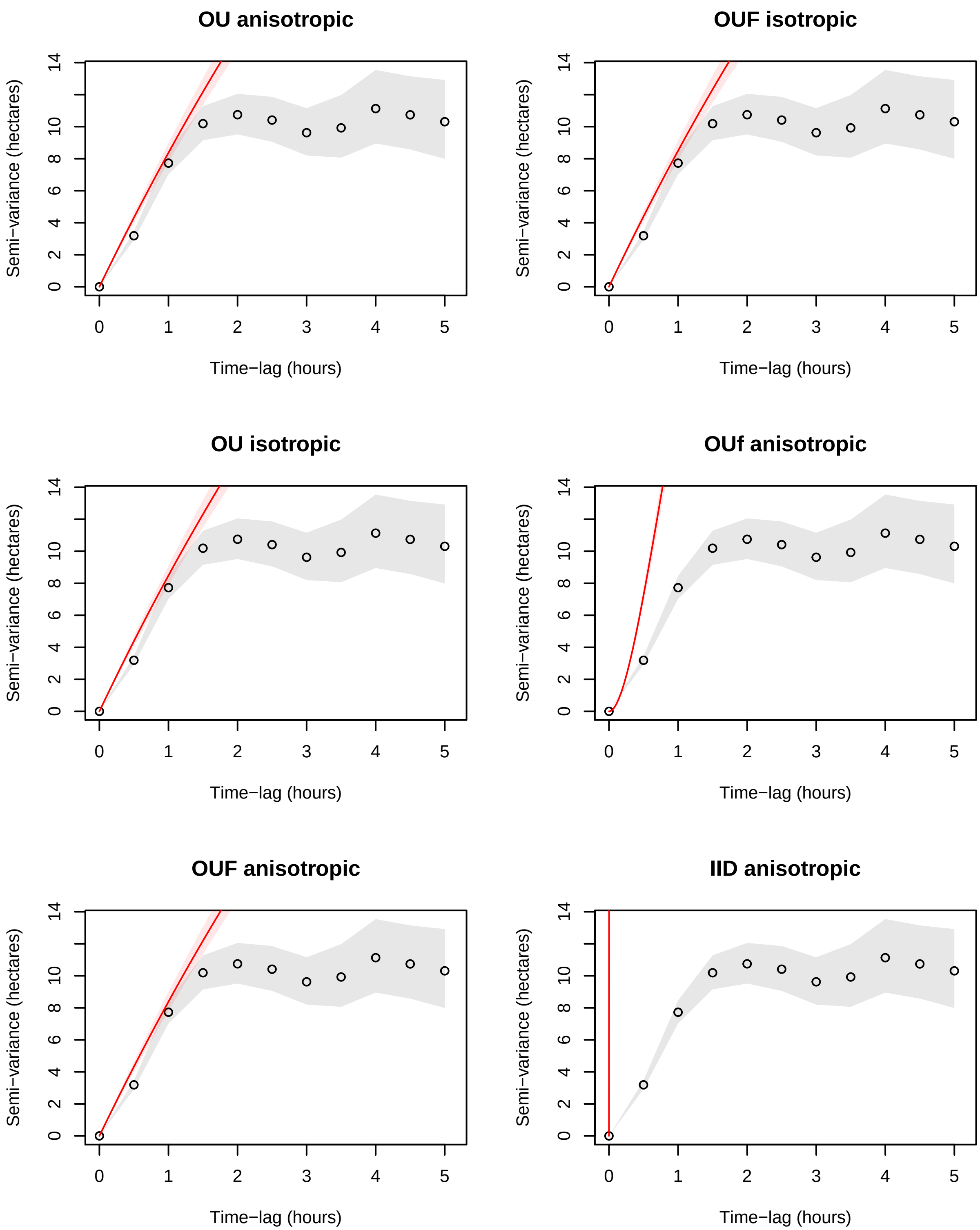

```
plot(svf, CTMM = OUa, col.CTMM = '#1b9e77', fraction = 0.1)
```



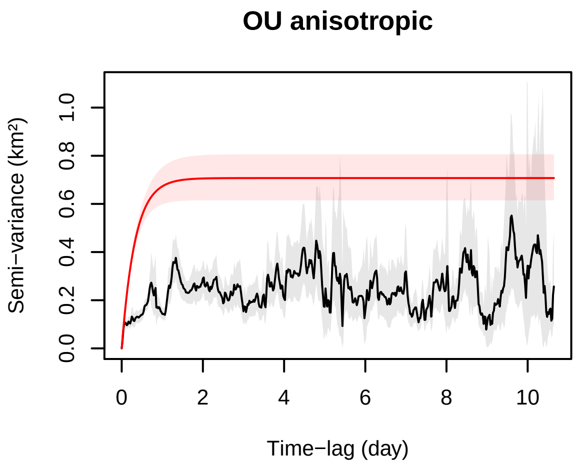

```r
plot(svf, CTMM = OUa, col.CTMM = '#1b9e77', fraction = 0.5)
```

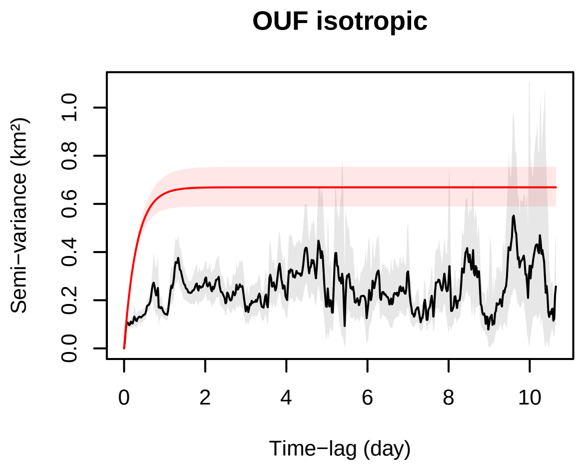


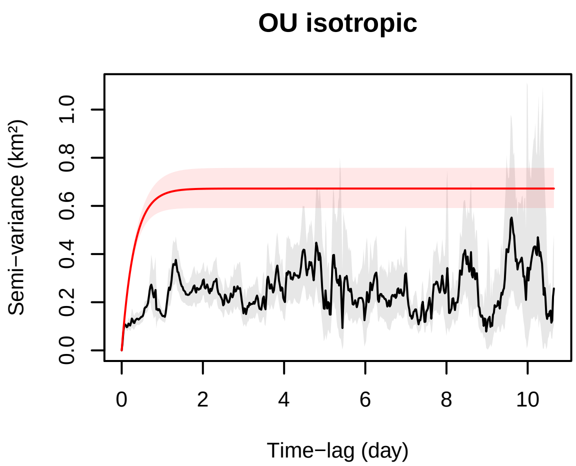
```r
layout(matrix(1:6, ncol = 2))
for(i in 1:6) plot(svf, CTMM = fitted.mods[[i]], col.CTMM = 'red',
                   fraction = 2e-4, main = names(fitted.mods)[i])
```

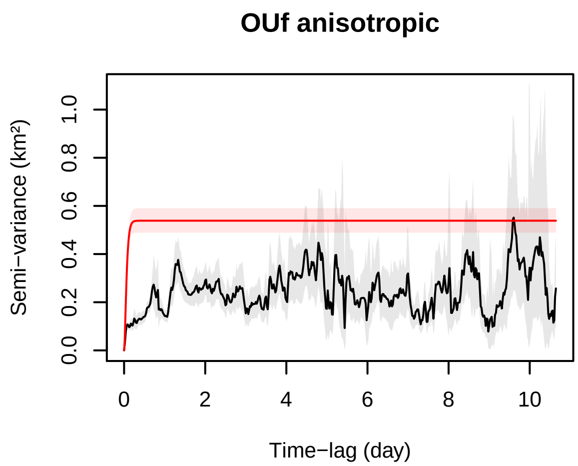


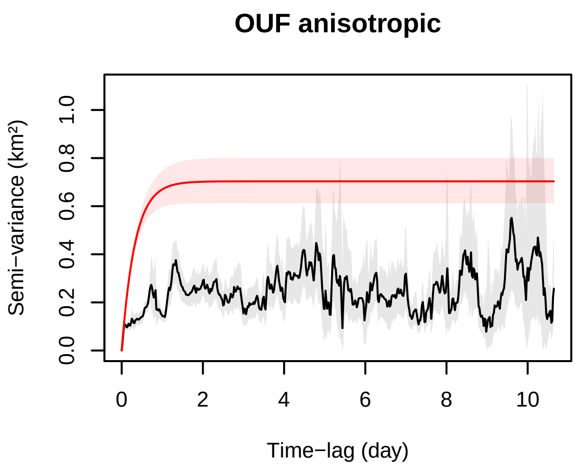
```r
for(i in 1:6) plot(svf, CTMM = fitted.mods[[i]], col.CTMM = 'red',
                   fraction = 0.01, main = names(fitted.mods)[i])
```

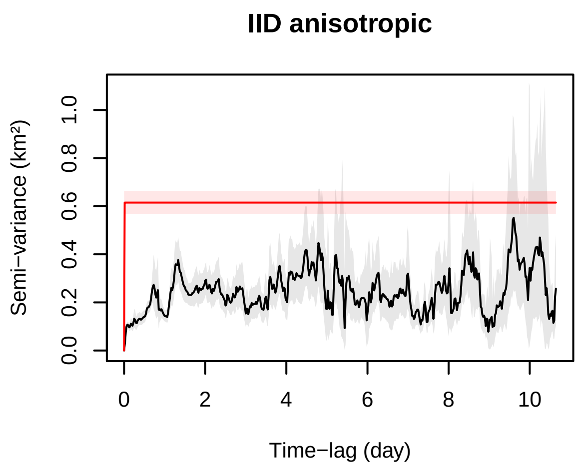




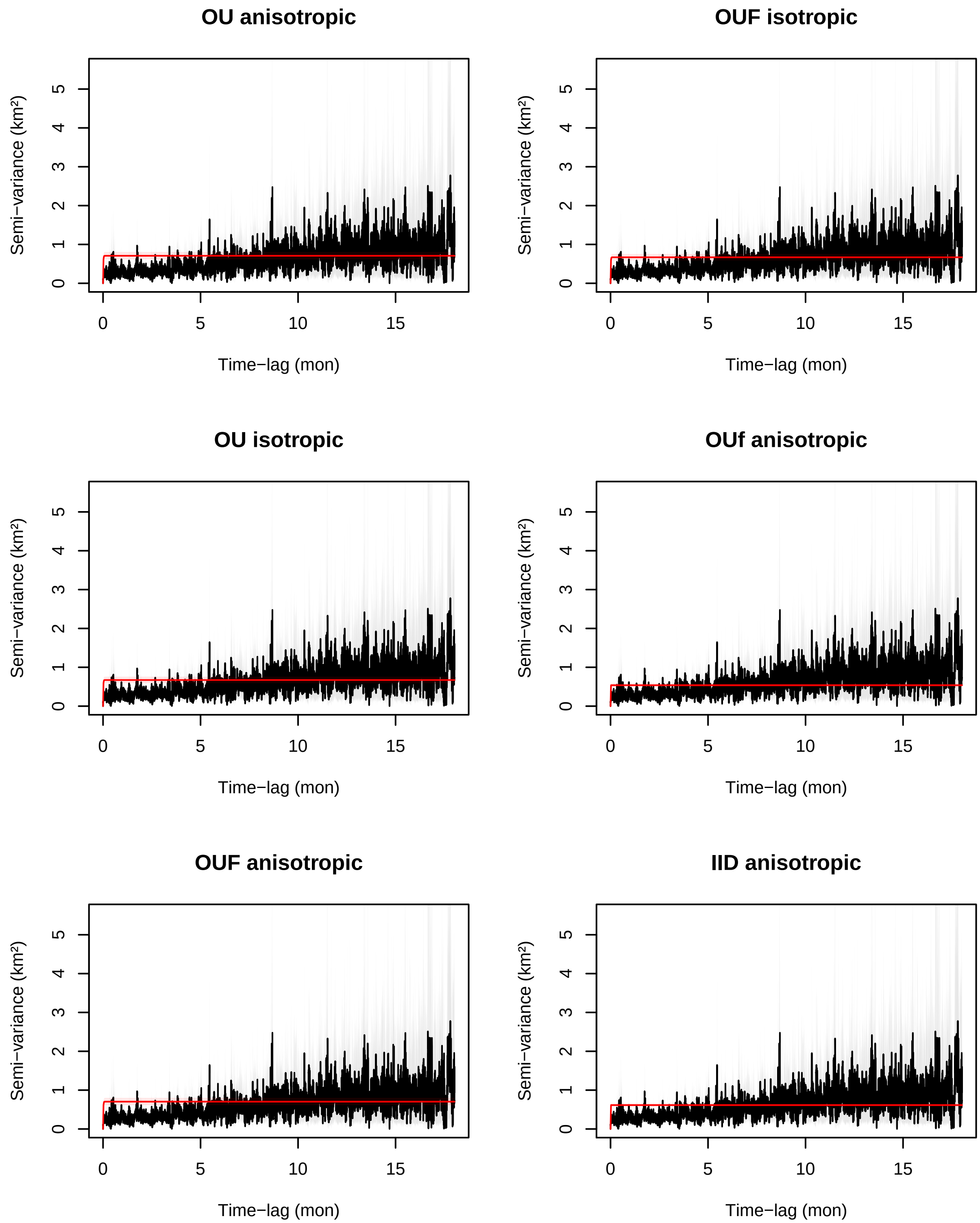




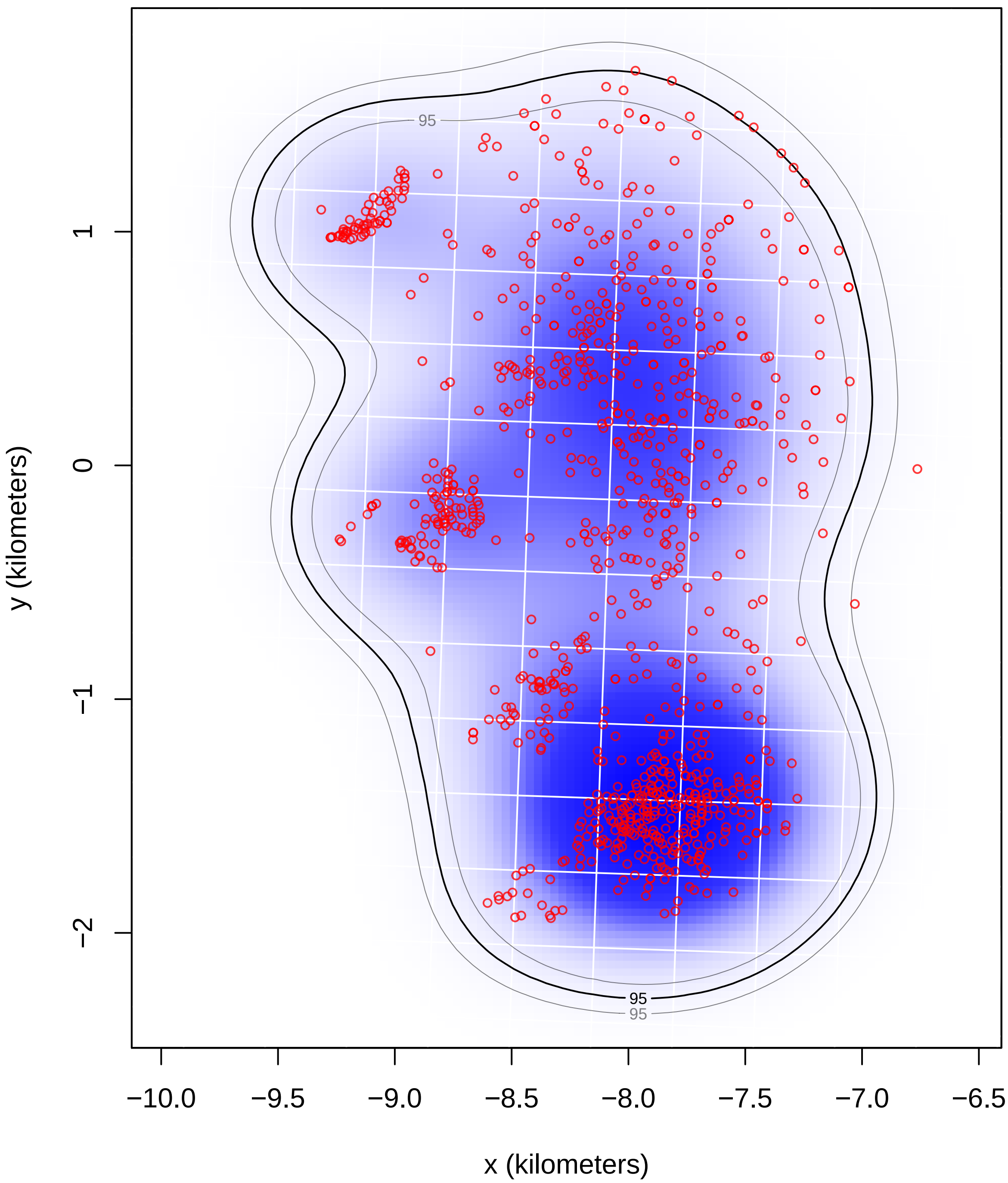




```
for(i in 1:6) plot(svf, CTMM = fitted.mods[[i]], col.CTMM = 'red',
                   fraction = 0.5, main = names(fitted.mods)[i])
```



```r
layout(1)
summary(OUa) # parameter and estimates with CIs
# $name
# [1] "OU anisotropic"
#
# $DOF
#     mean     area    speed
# 182.3084 216.7769   0.0000
#
# $CI
#                                low       est     high
# area (square kilometers) 11.558513 13.265792 15.08896
# t[position] (hours)       6.692213  7.939544  9.41936
speed(OUa) # OU model cannot make speed estimates (OUF can)
# Warning in speed.ctmm(OUa): Movement model is fractal.
#                        low est high
# speed (meters/second)   0 Inf  Inf
est <- akde(joana, CTMM = OUa)

plot(est) # 95% quantile of home range distribution with 95% CIs
plot(joana, add = TRUE) # datapoints
```


95
95
95
1
0
−1
−2
y (kilometers)
−10.0
−9.5
−9.0
−8.5
−8.0
−7.5
−7.0
−6.5
x (kilometers)