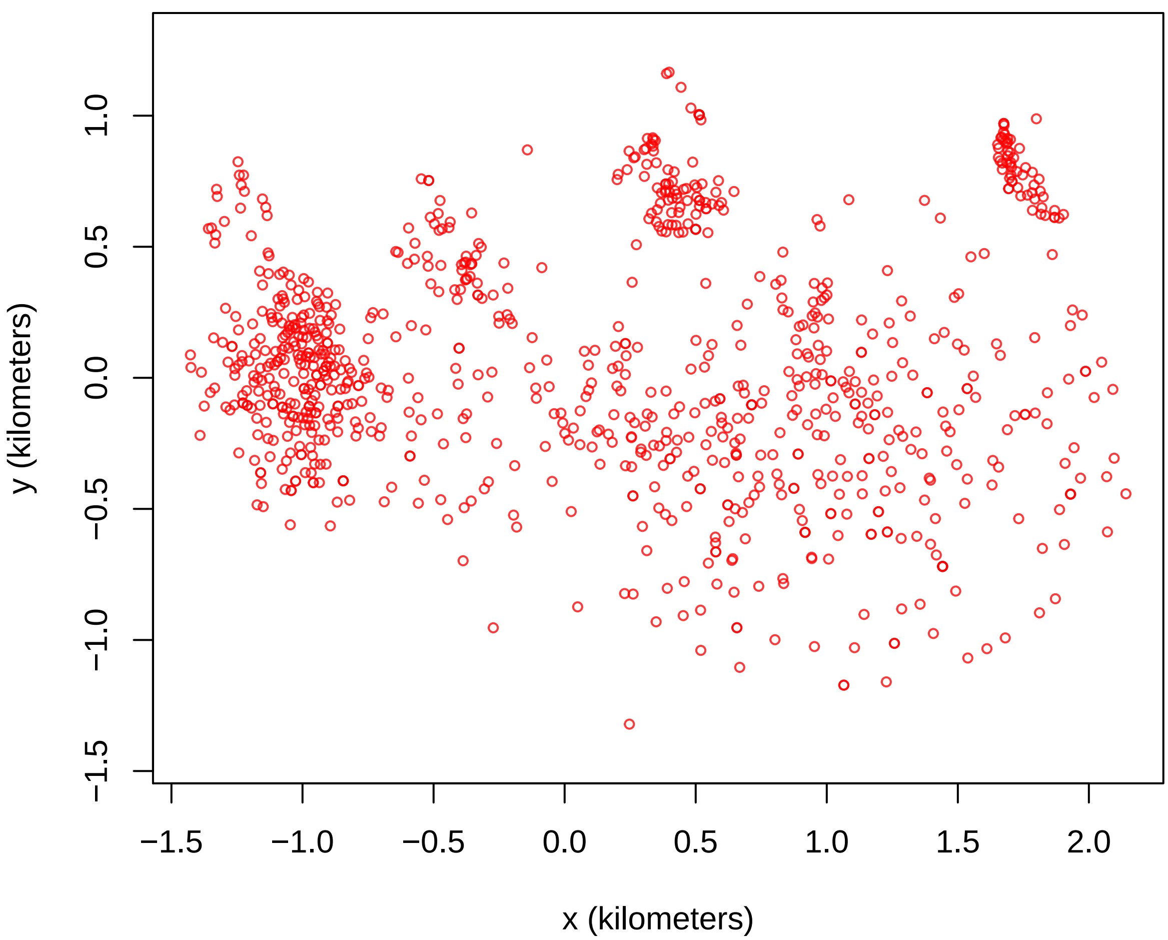
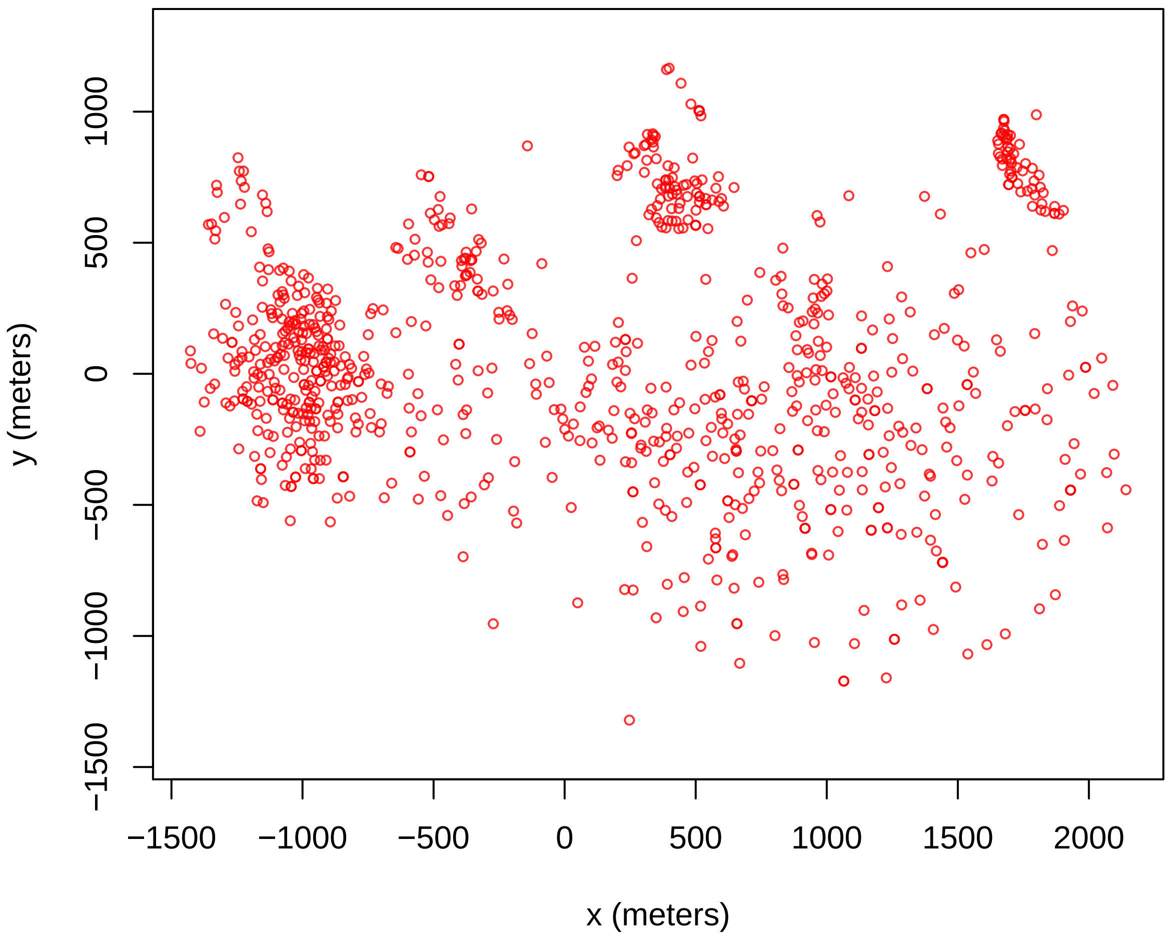
5 main steps to the analysis:

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

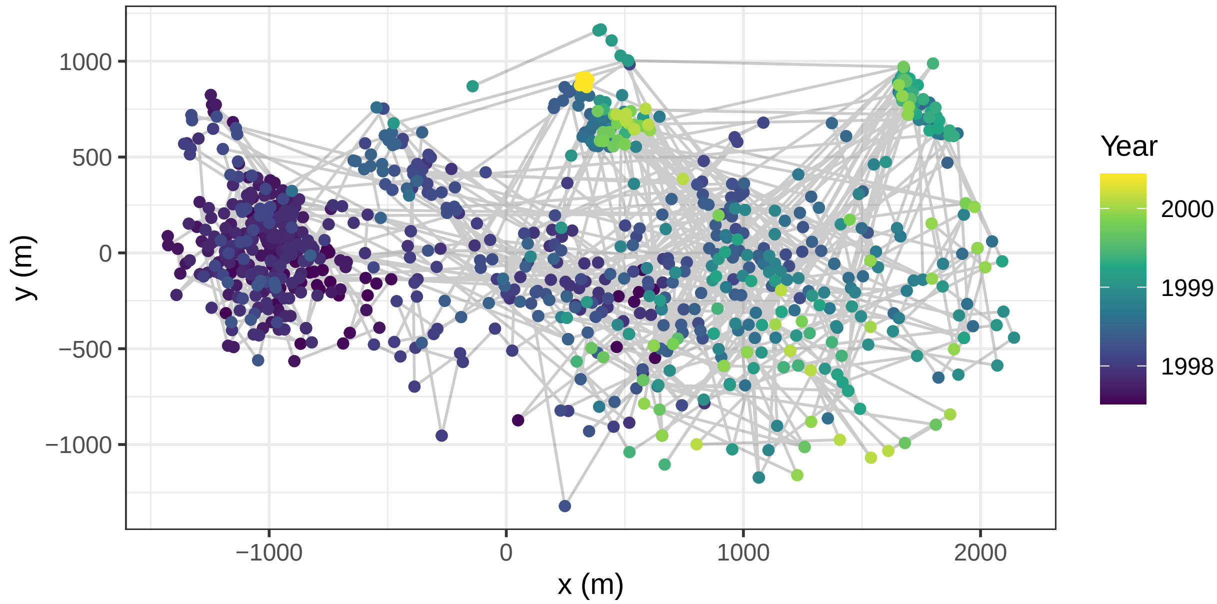
```
library('ctmm') # using the github version (0.6.1)
library('dplyr')
library('lubridate')
library('ggplot2')
theme_set(theme_bw())

tapir <-
 read.csv('../data/1_ATLANTICFOREST_11.csv') %>%
  filter(individual.local.identifier == 'AF_01_JOANA') %>% # use first tapir
  as.telemetry(timeformat = '%Y-%m-%d %H:%M:%S')

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



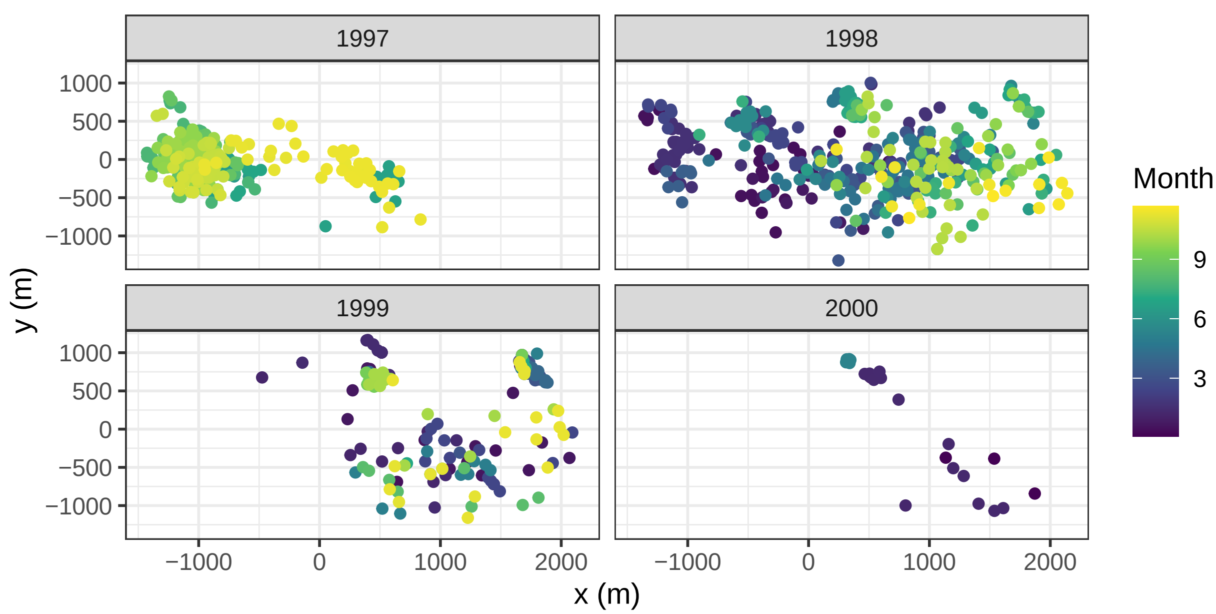
```r
plot(tapir, units = FALSE) # force SI units
```




```r
# eastward movement over time
ggplot(tapir, 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)')
```

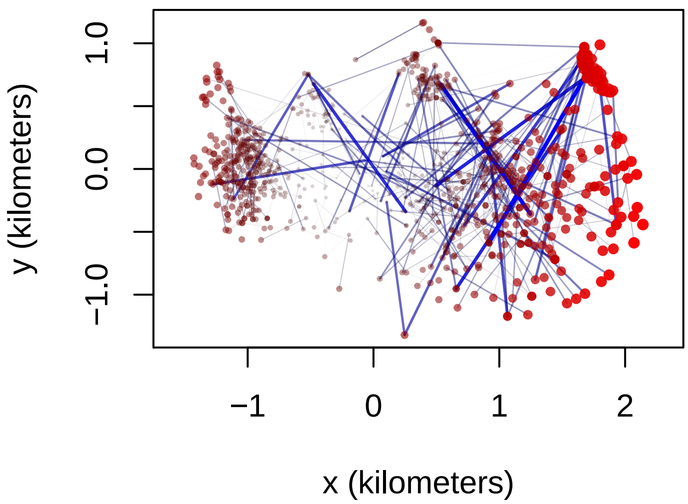



```
# facet plot to see seasonal detail
ggplot(tapir, aes(x, y)) +
  facet_wrap(~ year(timestamp)) +
  geom_point(aes(color = (decimal_date(timestamp) %% 1) * 12)) +
  scale_color_viridis_c('Month', breaks = c(3, 6, 9)) +
  labs(x = 'x (m)', y = 'y (m)')
```

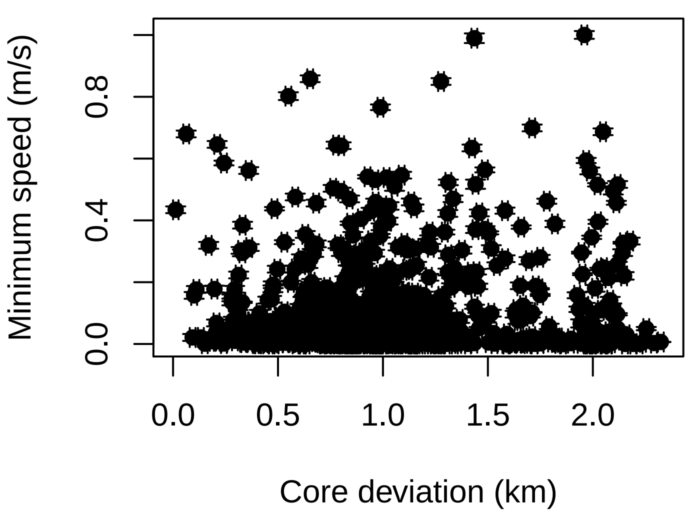


2. check for outliers

```
diagn <- outlie(tapir) # most red points are not actually unusual
```

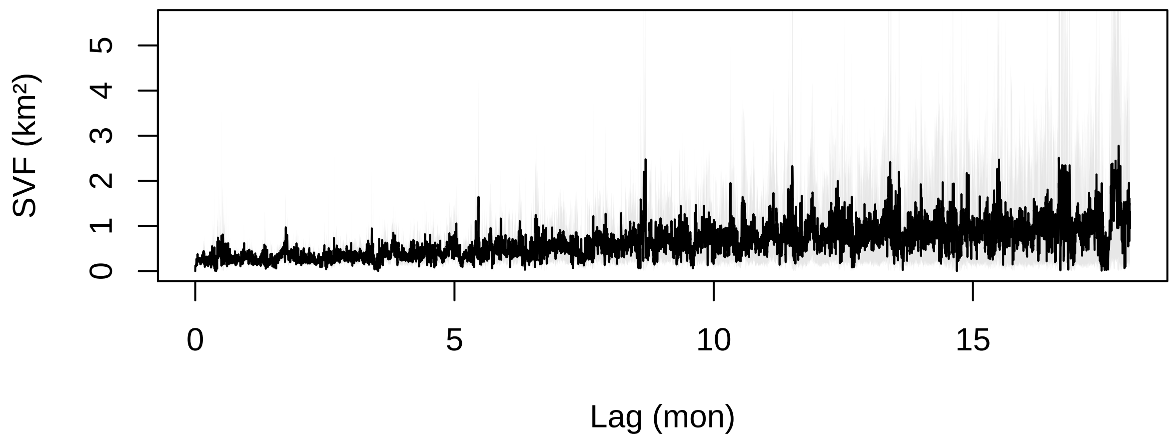



```
plot(diagn) # very narrow CIs, heterodskedastic in deviation and speed?
```

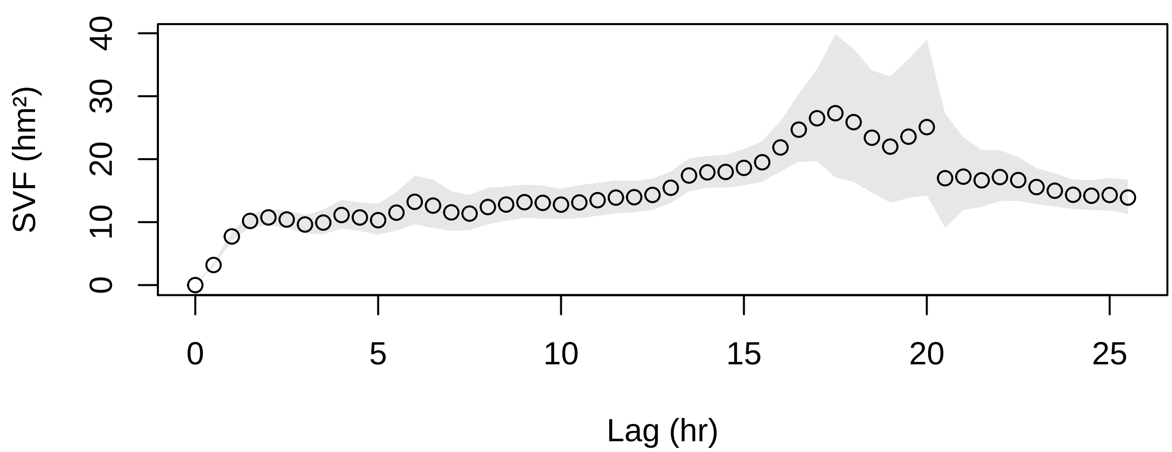


3. Variograms

```
svf <- variogram(tapir) # estimate semi-variance function
head(warnings(), n = 2)
# 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

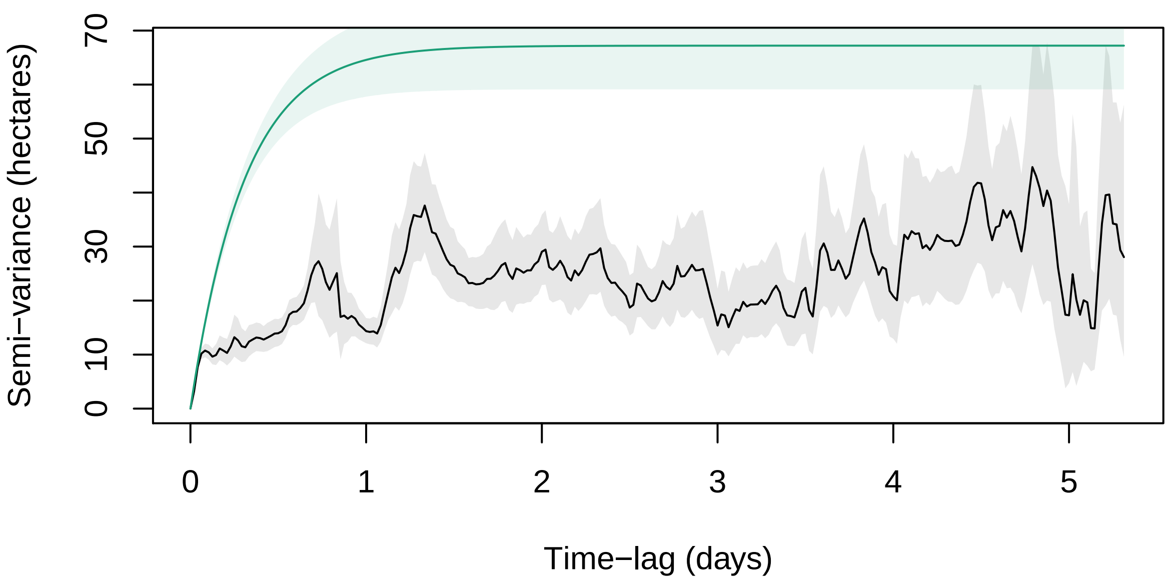
```
variogram.fit(svf, name = 'theta0', fraction = 0.0001)
fitted.mods <- ctmm.select(tapir, CTMM = theta0, verbose = TRUE, cores = 0)
```

Negligible difference between OU and OUF models (both anisotropic and isotropic), so choose the simplest model (OU isotropic?).

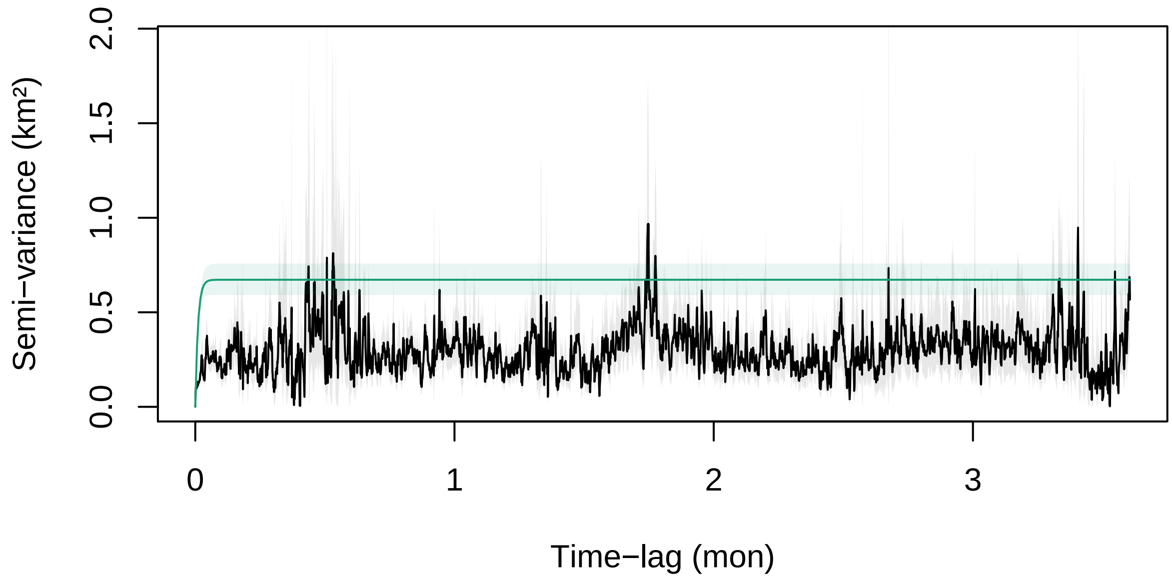
```
summary(fitted.mods)
#                          <U+0394>AICc        <U+0394>RMSPE (m) DOF[area]
# OU anisotropic      0.000000         153.35132  216.7769
# OU isotropic        1.480759         123.15327  248.7336
# OUF anisotropic     2.015385         149.91267  220.0352
# OUF isotropic       3.488707         120.41783  252.1144
# OUf anisotropic   465.039450           0.00000  408.7265
# IID anisotropic  1955.215930          70.34647  902.0004
OUi <- fitted.mods$`OU isotropic` # Ornstein-Uhlenbeck
```

5. calculate distance and speed estimates

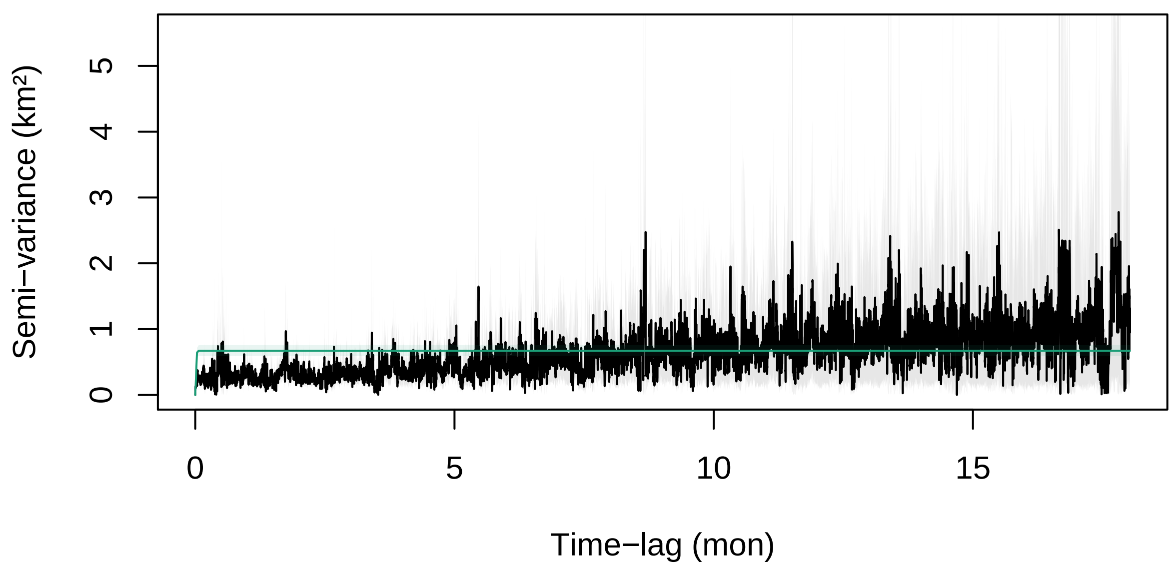
```
# plot semivariance function against empirical variogram
plot(svf, CTMM = OUi, col.CTMM = '#1b9e77', fraction = 0.005)
```



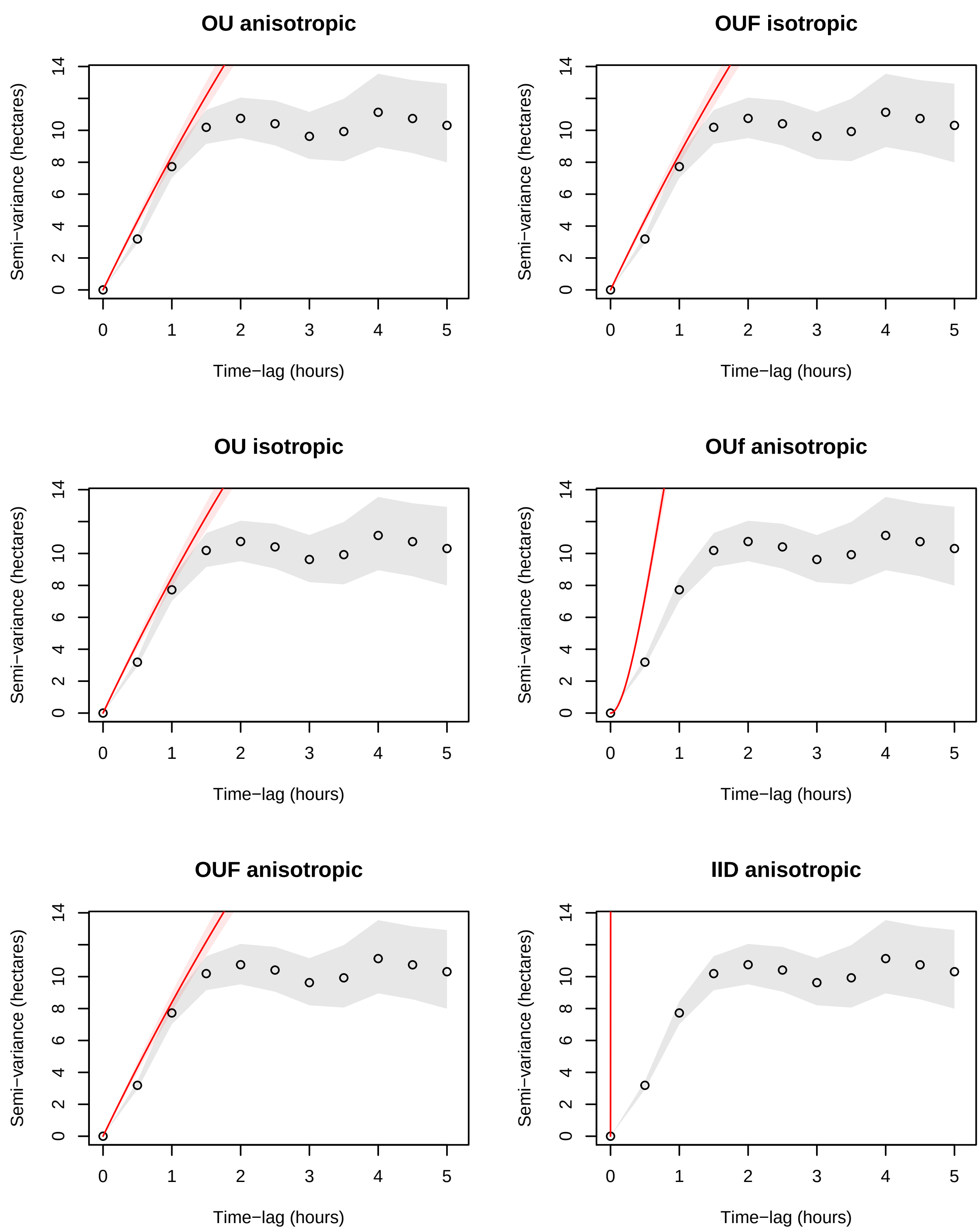

```
plot(svf, CTMM = OUi, col.CTMM = '#1b9e77', fraction = 0.1)
```



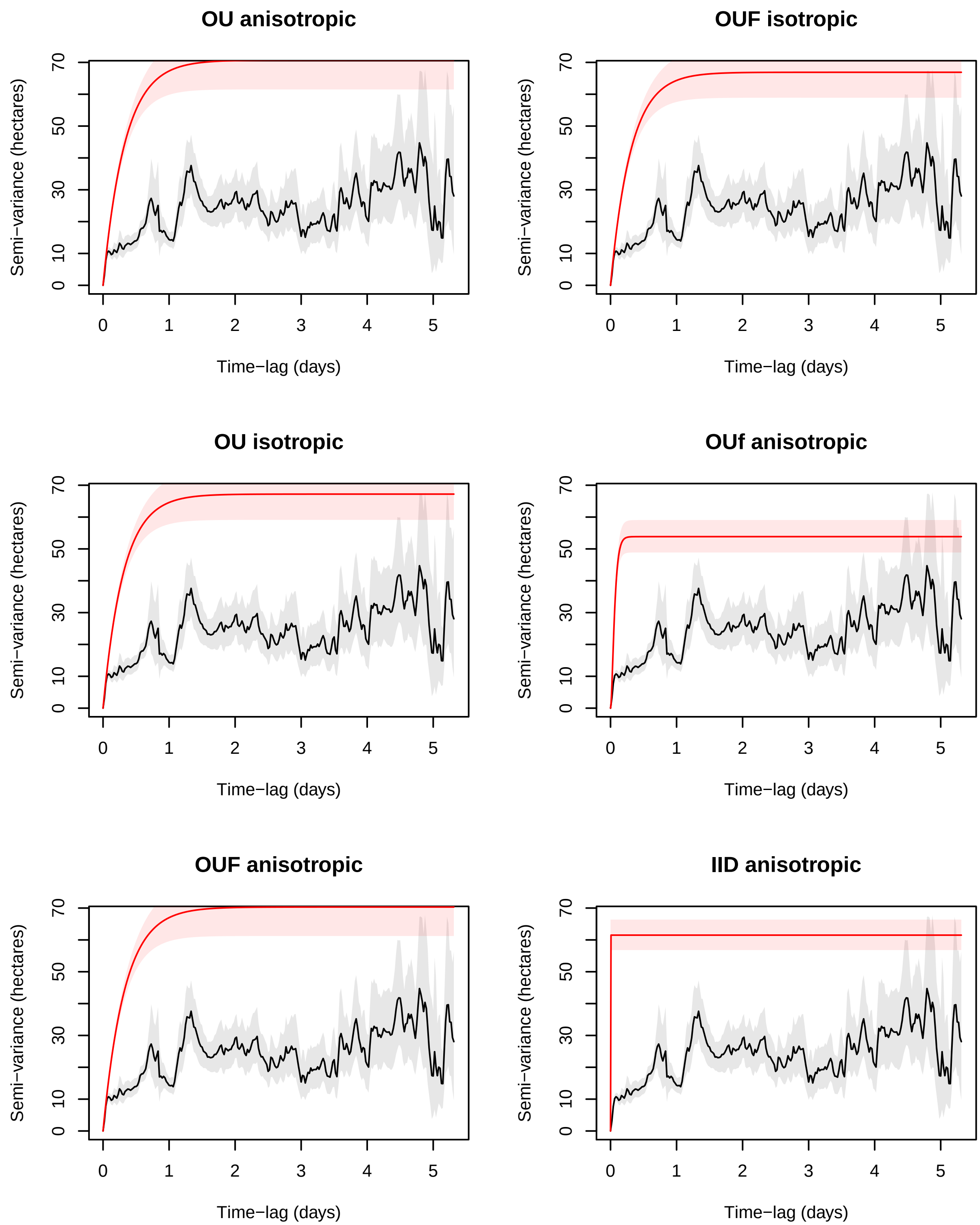

```
plot(svf, CTMM = OUi, col.CTMM = '#1b9e77', fraction = 0.5)
```



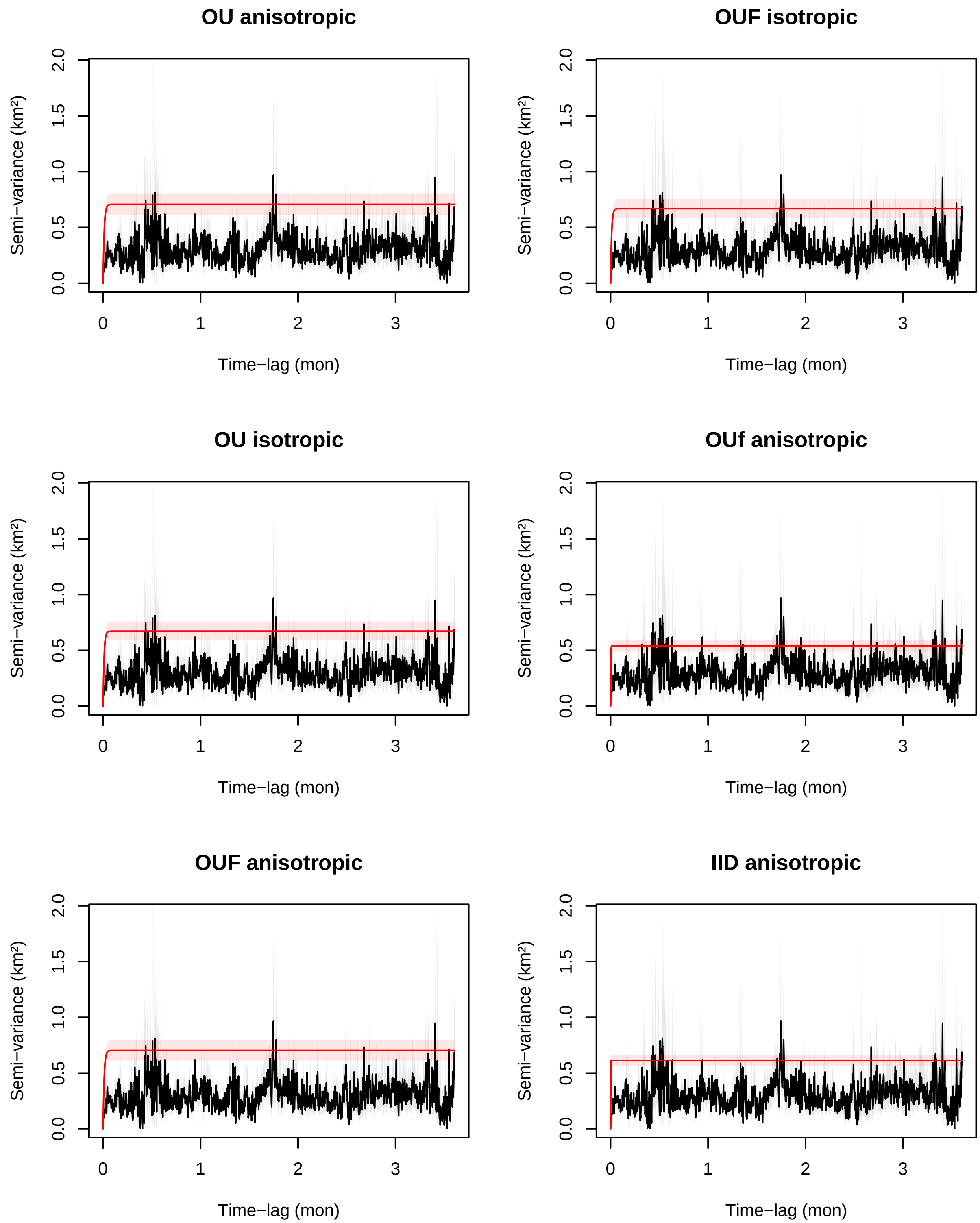
```r
layout(matrix(1:6, ncol = 2))
for(i in 1:6) plot(svf, CTMM = fitted.mods[[i]], col.CTMM = 'red',
                   fraction = 2e-4, main = names(fitted.mods)[i])
```



```r
for(i in 1:6) plot(svf, CTMM = fitted.mods[[i]], col.CTMM = 'red',
                   fraction = 0.005, main = names(fitted.mods)[i])
```



```r
for(i in 1:6) plot(svf, CTMM = fitted.mods[[i]], col.CTMM = 'red',
                   fraction = 0.1, main = names(fitted.mods)[i])
```



```
# parameter and estimates with CIs
summary(OUi)
# $name
# [1] "OU isotropic"
#
# $DOF
#     mean     area    speed
# 188.3007 248.7336   0.0000
#
# $CI
#                                low       est      high
# area (square kilometers) 11.126973 12.650376 14.270095
# t[position] (hours)       6.338129  7.410634  8.664622
summary(fitted.mods[[6]])
# $name
# [1] "IID anisotropic"
#
# $DOF
#     mean     area    speed
# 903.0000 902.0004   0.0000
#
# $CI
#                               low      est     high
# area (square kilometers) 8.207748 8.770867 9.352404
```