How resource abundance and stochasticity affect animals' space-use requirements

Appendix C: Empirical modeling

Contents

1 Modeling the tapir's movement over time **5**

2 Modeling E(R) and Var(R) over time **7**

3 Modeling the effects of E(R) and Var(R) on space use **15**

References **22**

This appendix illustrates the steps necessary to reproduce the tapir movement analysis and the related figure in the manuscript (fig. 5). The tapir data used here is from the work of Medici et al. (2022) and can be found at the GitHub repository located at https://github.com/StefanoMezzini/tapirs. Estimating the effects of resource abundance and unpredictability on the tapir's space-use requirements requires us to first estimate the changes in the tapir's space-use requirements (section 1) and the changes in resource abundance and variance experienced by the tapir (section 2) before we can estimate the relationship between resource dynamics and spatial use (section 3).

To minimize the computational costs of creating this appendix, we load the necessary objects through hidden `R` chunks rather than re-running all the code. Still, those interested in replicating the analyses can do so by using the code in the pdf document or the related `R` Markdown (`Rmd`) document (as well as the `R` scripts). All the packages and source scripts required to run the analyses in this document are listed in the code chunk below. For spatial data, we use the `MODIStsp` package (version 2.0.9, Busetto and Ranghetti 2016) to download the NDVI rasters, the `terra` package (version 1.7-39, Hijmans 2023) to work with the NDVI rasters, and the `sf` package (version 1.0-8, Pebesma 2018). We use the `dplyr` (version 1.0.10, Wickham et al. 2022), `purrr` (version 0.3.5, Henry and Wickham 2022), and `tidyr` (version 1.2.1, Wickham and Girlich 2022) packages for data wrangling and the `lubridate` package (version 1.8.0, Grolemund and Wickham 2011) for converting calendar dates to decimal dates. Finally, we used the `ctmm` package (version 1.1.0, Fleming and Calabrese 2021) and the `mgcv` package (version 1.8-41, Wood 2017) for modeling and the `ggplot2` (version 3.4.0, Wickham 2016) and `cowplot` (version 1.1.1, Wilke 2020) packages for plotting.

We start by attaching all necessary packages and custom functions we need:

```r
getwd() # change working directory or file paths if needed
```

```
[1] "H:/GitHub/hr-resource-stoch/writing"
```

```r
# attach all necessary packages
library('terra')    # to import and save rasters
library('dplyr')     # for data wrangling
library('purrr')     # for functional programming
library('tidyr')     # for data wrangling
library('ggplot2')   # for fancy plots
library('cowplot')   # for fancy multi-panel plots
library('ctmm')      # for movement modeling
library('mgcv')      # for empirical Bayesian GAMs
library('scam')      # for shape-constrained splines
library('lubridate') # for smoother date wrangling
library('sf')        # for spatial features
library('MODIStsp')  # for downloading NDVI rasters
theme_set(theme_bw()) # change default theme

# attach the beta location-scale family written by Simon Wood
source('../functions/betals.r')

# source all necessary scripts
source('../analysis/figures/default-figure-styling.R') # for color palettes
source('../earthdata-login-info.R') # personal login info for EarthData
source('../functions/window_hr.R') # function to calculate HRs
```

1 Modeling the tapir's movement over time

The script `analysis/tapir/tapirs-moving-window.R` estimates the seven-day spatial use of various tapirs from the Brazilian Cerrado. Here, we simplified the code so that it only estimates the spatial use of the tapir in the manuscript, Anna:

```
# import tapir data from https://github.com/StefanoMezzini/tapirs
anna <- readRDS('../../tapirs/models/tapirs-final.rds') %>%
  filter(name.short == 'ANNA')
anna_tel <- anna$data[[1]] # telemetry data

# re-project using the appropriate UTM projection for the Brazilian Cerrado
ctmm::projection(anna_tel) <- '+proj=utm +zone=22 +datum=NAD83 +units=m'

# calculate the 7-day home range estimate
window_hr(
  tel = anna_tel,
  window = 7 %#% 'day', # 1 week of data for sufficient sample size
  dt = 1 %#% 'day', # move window over by a single day each time
  fig_path = 'figures',
  rds_path = 'models')
anna_mw <-readRDS('../models/tapirs/CE_31_ANNA-window-7-days-dt-1-days.rds')
anna_mw
```

```
# A tibble: 457 x 15
       t_start     t_end dataset         guess  model  akde   hr_est_50 hr_lwr_50
         <dbl>     <dbl> <list>          <list> <list> <lis>      <dbl>     <dbl>
 1 1498285500     1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.555     0.362
 2 1498371900.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.418     0.280
 3 1498458300.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.482     0.337
 4 1498544700.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.597     0.403
 5 1498631100.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.566     0.382
 6 1498717500.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.708     0.459
 7 1498803900.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.642     0.427
 8 1498890300.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.758     0.492
 9 1498976700.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.814     0.534
10 1499063100.    1.50e9 <telemtry[,18]> <ctmm> <ctmm> <UD>       0.895     0.574
# i 447 more rows
# i 7 more variables: hr_upr_50 <dbl>, hr_est_95 <dbl>, hr_lwr_95 <dbl>,
#   hr_upr_95 <dbl>, t_center <dbl>, posixct <dttm>, date <date>
```

The `window_hr()` function estimates the tapir's home range using a sliding window approach with a window size of 7 days (using `ctmm`'s `%#%` operator for unit conversions) that starts with the set of days from 2017-06-23 to 2017-06-30 (extremes included), then shifts by one day (`dt = 1 %#% 'day'`), repeats the analysis for the next seven-day set (2017-06-24 to 2017-07-01), and continues doing so until it reaches the last set of days, 2018-09-22 to 2018-09-29. For each set of days, it fits a positional variogram, a continuous-time movement model (Fleming and Calabrese 2021), and a utilization distribution via autocorrelated kernel density estimation (Noonan et al. 2019; Silva et al. 2022). Finally, it saves an exploratory figure (fig. C1) to the `figures` folder and the tibble of times, telemetries, movement models, utilization distributions, and home range estimates (with 95% confidence intervals) to the `models` folder.

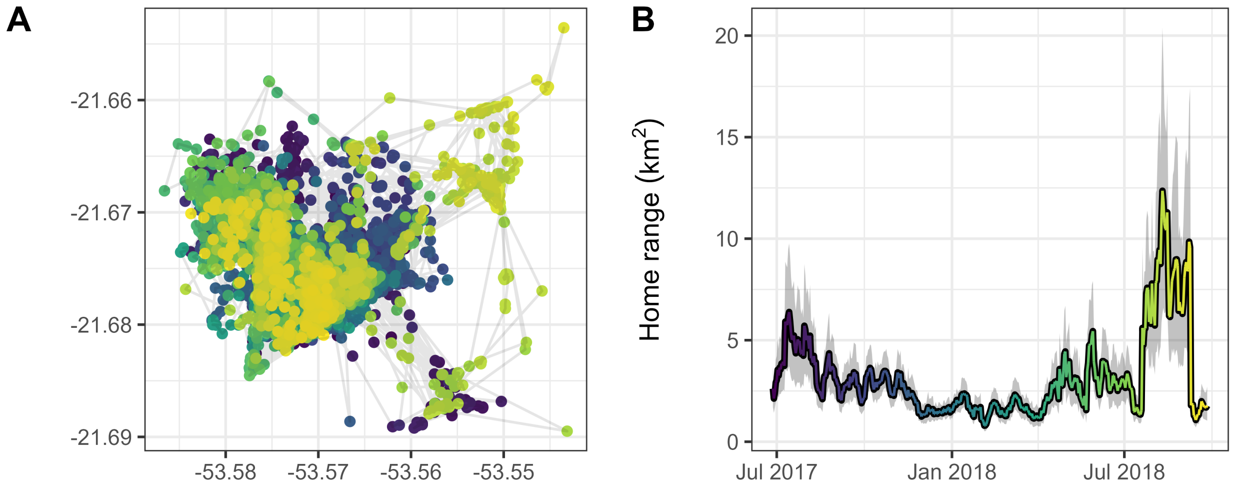


Figure C1: Exploratory figure created by the `window_hr`() function. Panel **A** shows the tapir's GPS locations, while panel **B** shows the seven-day home range estimates (95% utilization quantile) with 95% confidence intervals.

2 Modeling $\mathbf{E}(R)$ and $\mathbf{Var}(R)$ over time

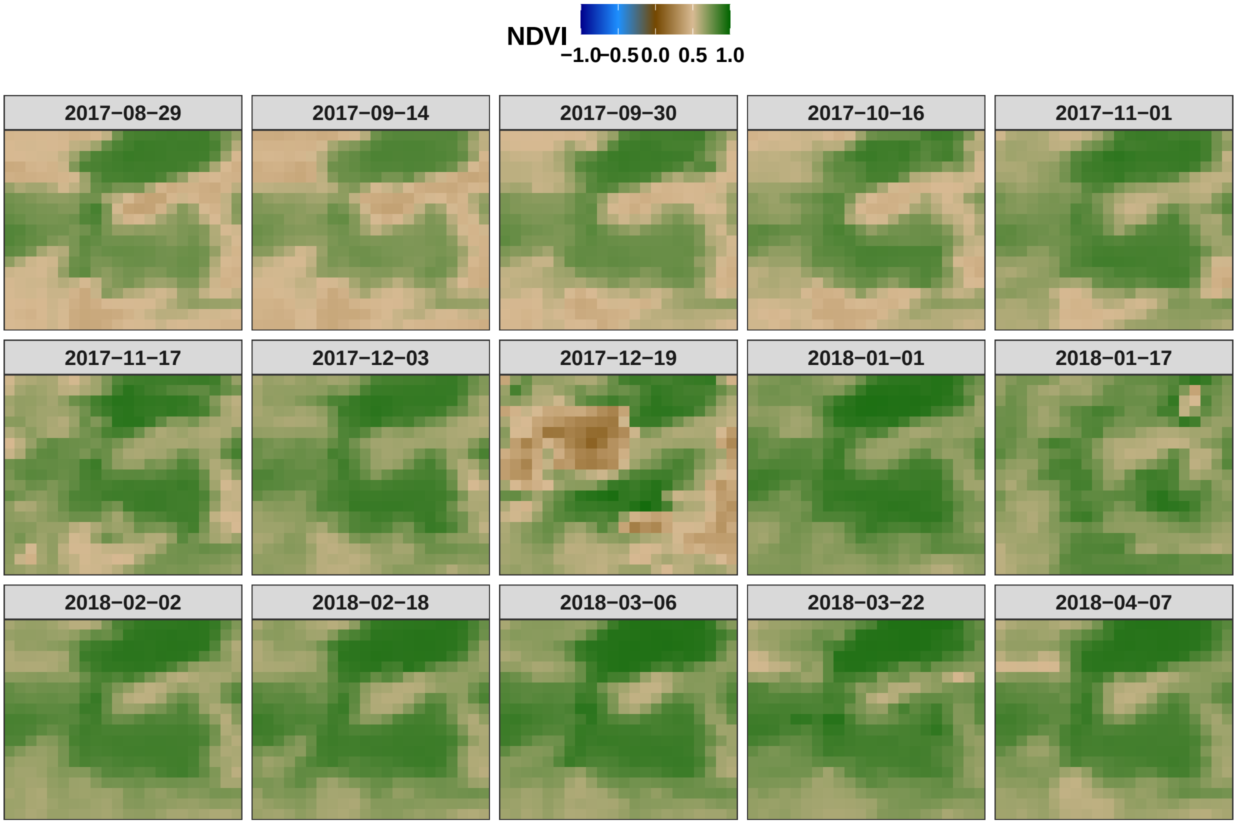
To estimate the resources in the tapir's habitat, we used satellite-measured Normalized Difference Vegetation Index (NDVI, see Pettorelli et al. 2005; Pettorelli et al. 2011). We downloaded the data using the `MODIStsp` `R` package with the following code:

```r
anna_ud <- anna$akde[[1]] # extract the tapir's utilization distribution

bbox <-
  SpatialPolygonsDataFrame.UD(anna_ud, # convert to a spatial object
                              level.UD = 0.9995, # utilization quantile
                              level = 0) %>% # no CIs
  st_as_sf() %>%
  st_transform(crs = '+proj=longlat') %>%
  st_bbox()

# download NDVI rasters (if needed, create all necessary folders first)
MODIStsp(gui = FALSE, # do not use the browser GUI, only run in R
         out_folder = 'data/ndvi-rasters/tapir-anna',
         selprod = 'Vegetation Indexes_16Days_250m (M*D13Q1)',
         prod_version = '061', # 2022 raster version
         bandsel = 'NDVI', # NDVI layer only
         sensor = 'Terra', # only terrestrial values, ignore water
         user = USERNAME, # Earthdata username for urs.earthdata.nasa.gov
         password = PASSWORD, # your Earthdata password
         start_date = format(min(anna_tel$timestamp) - 16, '%Y.%m.%d'),
         end_date = format(max(anna_tel$timestamp) + 16, '%Y.%m.%d'),
         spatmeth = 'bbox', # use a bounding box for the extent
         bbox = bbox, # spatial file for raster extent
         out_projsel = 'User Defined', # use specified projection
         output_proj = '+proj=longlat', # download unprojected raster
         resampling = 'bilinear', # raster resampling method for new proj
         delete_hdf = TRUE, # delete HDF files after download is complete
         scale_val = TRUE, # convert from integers to floats within [-1, 1]
         out_format = 'GTiff', # output format
         verbose = TRUE) # print processing messages

# save NDVI data as an rds file of a tibble
list.files(path = 'data/ndvi-rasters/tapir-anna/VI_16Days_250m_v61/NDVI/',
           pattern = '.tif', full.names = TRUE) %>%
  rast() %>% # import all rasters as a single stack
  as.data.frame(xy = TRUE) %>% # convert to a data frame
  pivot_longer(-c(x, y)) %>% # change to long format (x, y, name, value)
  transmute(long = x, # rename x column
            lat = y, # rename y column
            date = substr(name, # change name to a date
                          start = nchar('MOD13Q1_NDVI_x'),
                          stop = nchar(name)) %>%
              as.Date(format = '%Y_%j'), # format is year_julian date
            ndvi = value, # rename value column
            dec_date = decimal_date(date)) %>%
  saveRDS('data/ndvi-rasters/tapir-anna/tapir-anna-data.rds')

# import NDVI data
anna_ndvi <-
  readRDS('data/ndvi-rasters/tapir-anna/tapir-anna-data.rds') %>%
  mutate(dec_date = decimal_date(date))
anna_ndvi
```

```
# A tibble: 13,376 x 5
    long   lat date          ndvi dec_date
   <dbl> <dbl> <date>     <dbl>    <dbl>
 1 -53.6 -21.7 2017-06-10 0.626    2017.
 2 -53.6 -21.7 2017-06-26 0.595    2017.
 3 -53.6 -21.7 2017-07-12 0.469    2018.
 4 -53.6 -21.7 2017-07-28 0.421    2018.
 5 -53.6 -21.7 2017-08-13 0.426    2018.
 6 -53.6 -21.7 2017-08-29 0.479    2018.
 7 -53.6 -21.7 2017-09-14 0.440    2018.
 8 -53.6 -21.7 2017-09-30 0.488    2018.
 9 -53.6 -21.7 2017-10-16 0.468    2018.
10 -53.6 -21.7 2017-11-01 0.524    2018.
# i 13,366 more rows
```

We removed the raster for 2017-12-19 because the values were unusually low for the region. We hypothesize the change in NDVI was drastic, temporary, and widespread because of a sudden flood (which is common for the Cerrado):

```
anna_ndvi %>%
  filter(date >= as.Date('2017-08-29'), date <= as.Date('2018-04-07')) %>%
  ggplot() +
  facet_wrap(~ date, nrow = 3) + # a raster for each date
  coord_equal() + # keep the scaling of x and y equal
  geom_tile(aes(long, lat, fill = ndvi)) +
  scale_x_continuous(NULL, breaks = NULL, expand = c(0, 0)) +
  scale_y_continuous(NULL, breaks = NULL, expand = c(0, 0)) +
  scale_fill_gradientn('NDVI', colours = ndvi_pal, limits = c(-1, 1)) +
  theme(legend.position = 'top')
```



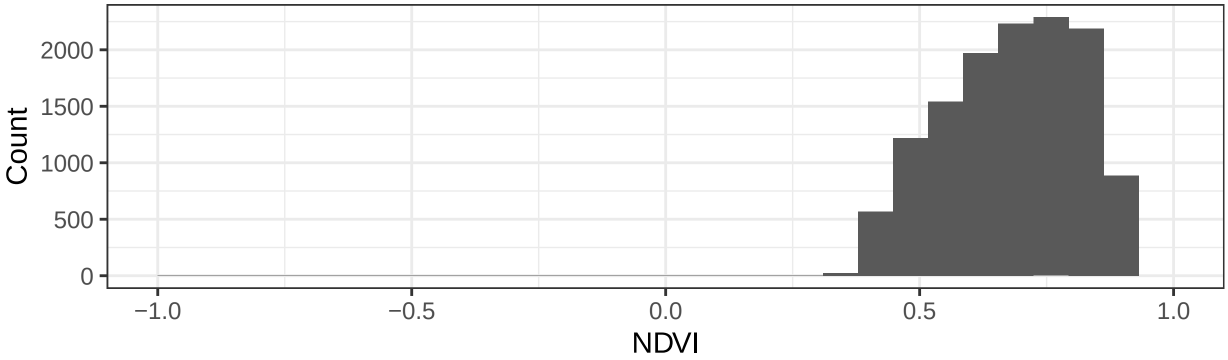
```
anna_ndvi <- filter(anna_ndvi, date != '2017-12-19') # remove bad values
```

Next, we estimate the mean and variance in NDVI using a Generalized Additive Model for location and scale (GAMLS: Stasinopoulos and Rigby 2007). Ideally, we would model NDVI using a family of distributions that accounts for the fact that NDVI cannot be less than -1 or greater than 1, but no such family is available in the `mgcv` package. However, since the NDVI data is far from 0 (see the histogram below), we decided to keep NDVI unscaled. This prevents the model from predicting values that correspond to water or snow (which are not expected to occur in the study area). Mathematically, this approach would be comparable to predicting in a Bayesian framework with a prior with zero probability for any NDVI values below zero. In environments where values below zero are probable, one could use the beta family after applying the linear transformation

$$y^* = \frac{y+1}{2},$$

where y is the original NDVI value (between -1 and 1) and y^* is the NDVI value scaled between 0 and 1.

We fit the beta GAMLS via the `mgcv` package (note `family = betals()` in the code chunk below). The `betals` family accepts a list of two predictors: one for the mean parameter, μ, and one for the scale parameter, ϕ, and it uses logit link functions for both parameters (see fig. C2). The variance of the distribution is a function of both parameters: $\sigma^2 = \mu(1-\mu)\phi$.



```
m_ndvi <-
  gam(list(
    # mean predictor
    ndvi ~ # not scaling because range is in (0, 1)
      s(long, lat, bs = 'ds', k = 20) + # mean over space
      s(dec_date, bs = 'tp', k = 20),   # high k to account for adaptions
    # scale predictor (sigma2 = mu * (1 - mu) * scale)
    ~
      s(long, lat, bs = 'ds', k = 10) + # scale over space
      s(dec_date, bs = 'tp', k = 10)),  # scale over time
    family = betals(),
    data = anna_ndvi,
    method = 'REML') # REstricted Maximum Likelihood
```

Note that when fitting location-scale GAMs, one should pay particular attention to the number of knots used for each smooth term. While using a penalized maximum likelihood method such as REML (`method = 'REML'`) helps avoid over-fitting the model, finding the right balance between each of the `k` values is crucial. Excessively smooth terms for the mean can inflate the scale term (fig. C3), while excessively wiggly terms for the mean can cause the scale to be under-estimated (fig. C4). Ultimately, each of the `k` values should be decided in such a way as to mimic the animal's responsiveness, adaptability, motility, and memory (or ability to predict cycles or events). If one is unsure where to start from, keeping the `k` for the scale terms below half the `k` for the mean terms is a good starting point.

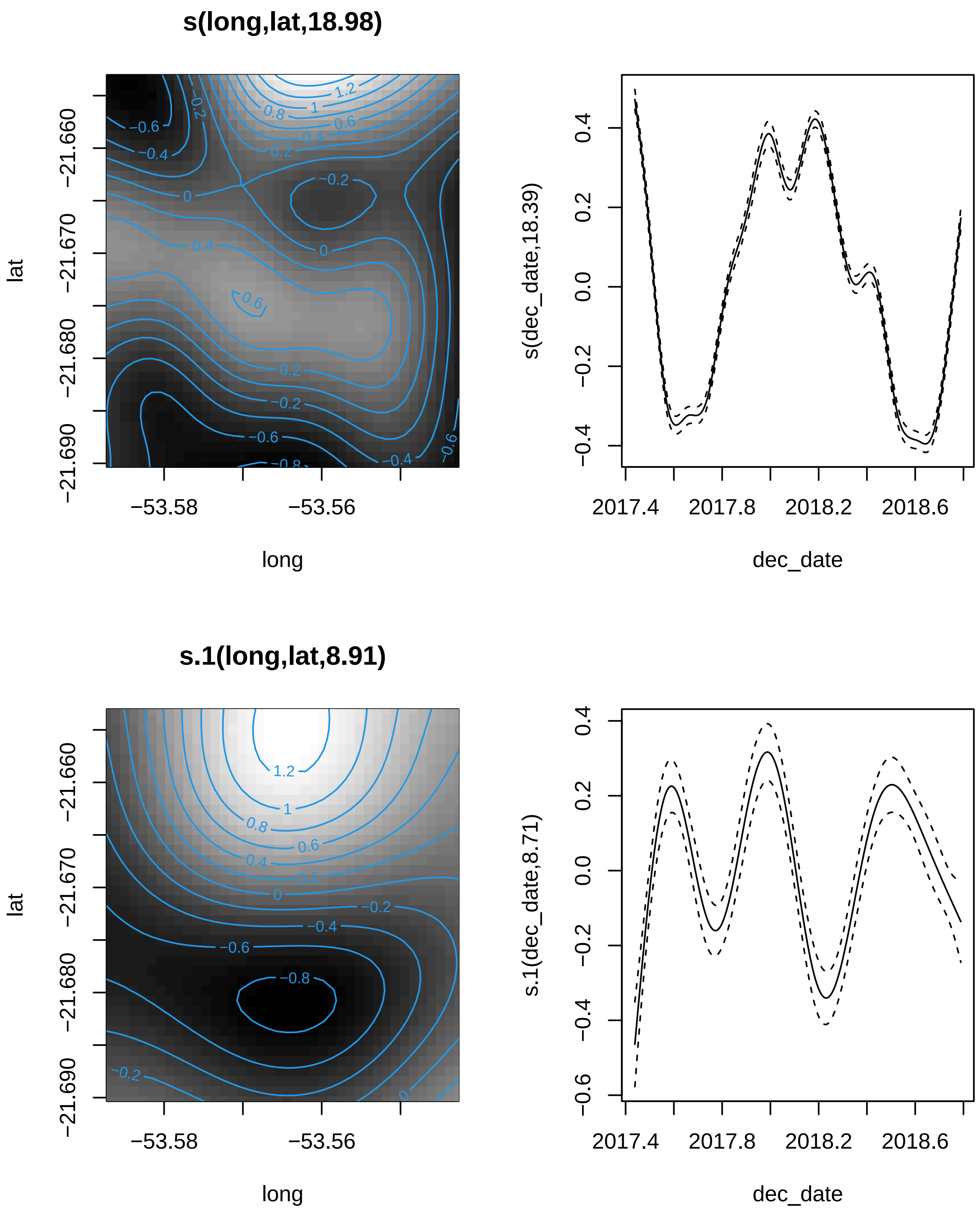


Figure C2: Estimated spatiotemporal trends in mean and scale parameters using the model detailed in the code chunk above. Estimates are provided on the logit link scale. The estimated degrees of freedom for each term can be seen in parentheses in the title of the spatial terms and the y-axis labels of the temporal terms. Dashed lines inticate the 95% credible intervals for the temporal terms.

```r
gam(list(
  # mean predictor (default ks are too small)
  ndvi ~ s(long, lat, bs = 'ds', k = 10) + s(dec_date, bs = 'tp', k = 10),
  # scale predictor (excessively high k)
  ~ s(long, lat, bs = 'ds', k = 20) + s(dec_date, bs = 'tp', k = 20)),
  family = betals(), data = anna_ndvi, method = 'REML') %>%
  plot(pages = 1, scheme = 3, n = 250, scale = 0)
```

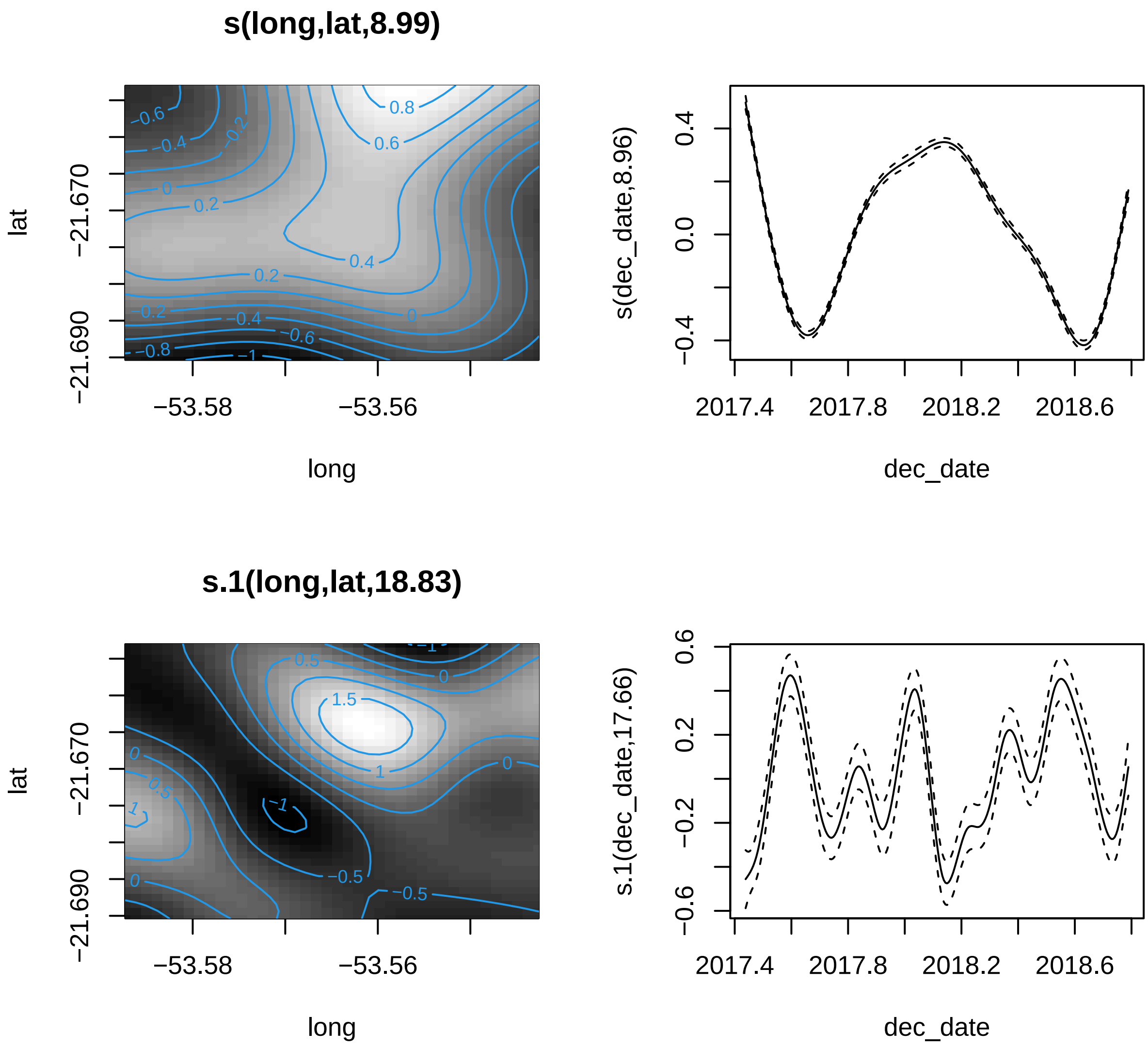


Figure C3: Estimated mean and scale parameters with excessively small basis size k for the mean's smooth terms and excessively large for the scale's smooth terms. The estimated degrees of freedom for each term can be seen in parentheses in the title of the spatial terms and the y-axis labels of the temporal terms. Dashed lines inticate the 95% credible intervals for the temporal terms.

```
gam(list(
  # mean predictor (ks too large)
  ndvi ~ s(long, lat, bs = 'ds', k = 50) +
    s(dec_date, bs = 'tp', k = n_distinct(anna_ndvi$dec_date)),
  # scale predictor (ks too small)
  ~ s(long, lat, bs = 'ds', k = 5) + s(dec_date, bs = 'tp', k = 5)),
  family = betals(), data = anna_ndvi, method = 'REML') %>%
  plot(pages = 1, scheme = 3, n = 250, scale = 0)
```

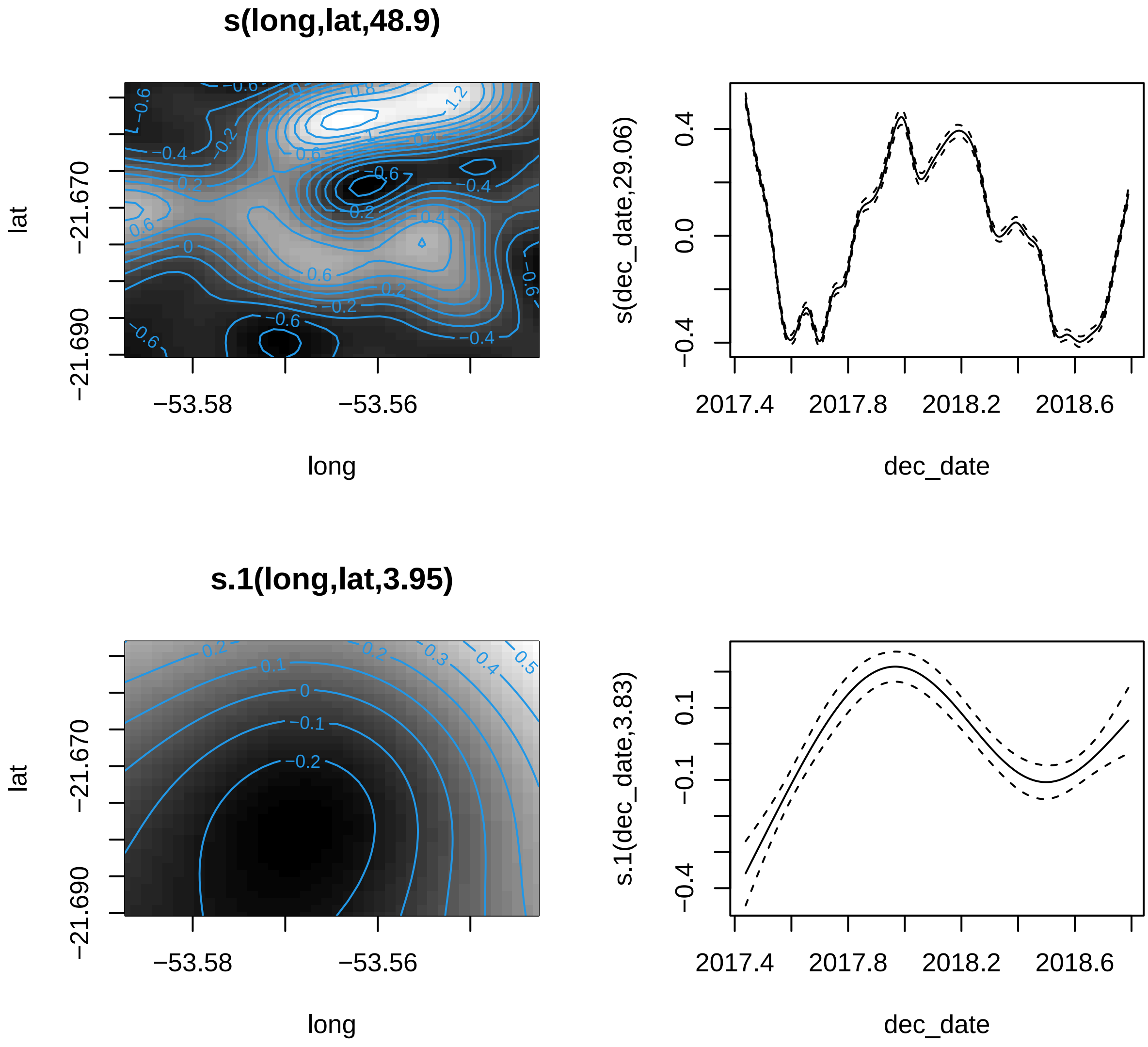


Figure C4: Estimated mean and scale parameters in NDVI with excessively large basis size `k` for the smooth term of time (`dec_date`) for the mean in NDVI and excessively low for the variance temporal smooth term. The estimated degrees of freedom for each term can be seen in parentheses in the title of the spatial terms and the y-axis labels of the temporal terms. Dashed lines inticate the 95% credible intervals for the temporal terms.

3 Modeling the effects of E(R) and Var(R) on space use

We start by predicting the mean and variance in NDVI experienced by the tapir in the known positions using the beta GAMLS.

```
anna_tel <-
  data.frame(anna_tel) %>% # convert telemetry to data frame
  rename(long = longitude, lat = latitude) %>%
  mutate(dec_date = decimal_date(timestamp)) %>% # needed for predictions
  bind_cols(., # bind telemetry to predictions
            predict(m_ndvi, newdata = ., type = 'response',
                    se.fit = FALSE) %>%
              data.frame() %>% # convert list of predictions to data frame
              # didn't scale NDVI to [0, 1], so no need to back-transform
              transmute(mu = X1, sigma2 = X1 * (1 - X1) * X2)) %>%
  as_tibble()
```

Next, we can estimate the mean and variance in NDVI for each 7-day period by taking the average for the GPS locations within each period. We also add a column of weights that are based on the home range confidence estimates' 95% intervals.

```
tapir <-
  readRDS('../models/tapirs/CE_31_ANNA-window-7-days-dt-1-days.rds') %>%
  mutate(est = map(dataset,
                   \(.d) filter(anna_tel, timestamp %in% .d$timestamp)),
         mu = map_dbl(est, \(.d) mean(.d$mu)),
         sigma2 = map_dbl(est, \(.d) mean(.d$sigma2))) %>%
  select(date, mu, sigma2, hr_lwr_95, hr_est_95, hr_upr_95) %>%
  mutate(weight = 1 / (hr_upr_95 - hr_lwr_95), # small CI => large weight
         weight = weight / mean(weight)) # standardize weights
sum(tapir$weight) == nrow(tapir) # ensure weights are right
```

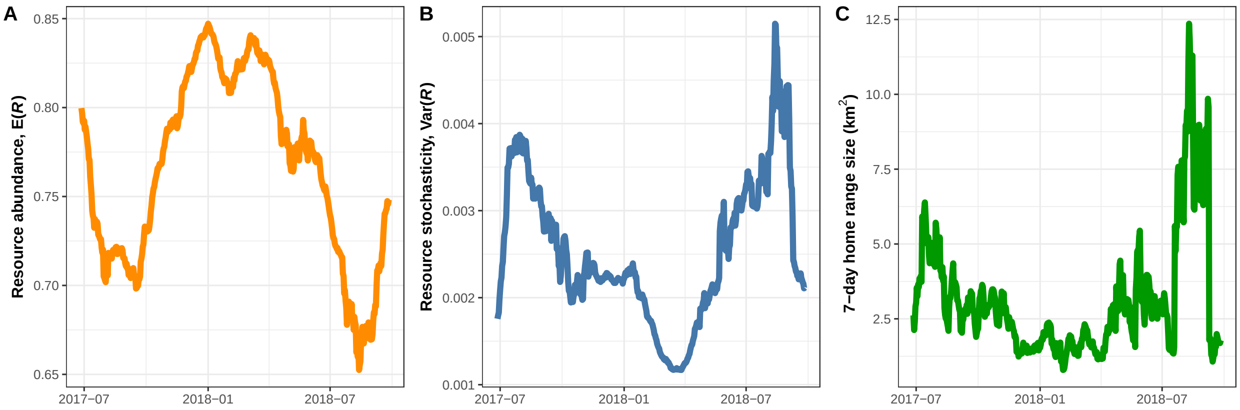
```
[1] TRUE
```

We now have all we need to create the left side of figure 5 from the main manuscript.

```r
theme_set(theme_bw()) # need to set the theme again for some reason
# axis labels (spaces in hr_lab to avoid overlap with the plot label)
# unicode characters removed because they knitting errors
e_r <- bquote(paste(bold('Resource abundance, E('), bolditalic(R),
                    bold(')')))
v_r <- bquote(paste(bold('Resource stochasticity, Var('), bolditalic(R),
                    bold(')')))
hr_lab <- bquote(bold(paste('7-day home range size (', km^2, ')    ')))

l_grobs <- lapply(
  list(ggplot(tapir, aes(date, mu)) + # mean
         geom_line(color = pal[1], size = 2) +
         labs(x = NULL, y = e_r),
       ggplot(tapir, aes(date, sigma2)) + # variance
         geom_line(color = pal[2], size = 2) +
         labs(x = NULL, y = v_r),
       ggplot(tapir, aes(date, hr_est_95)) + # 95% home range
         geom_line(color = pal[3], size = 2) +
         labs(x = NULL, y = hr_lab)),
  as_grob) # convert to grid graphical objects (grobs)
```

```
Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use ‘linewidth‘ instead.
This warning is displayed once every 8 hours.
Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
generated.
```

```r
# align left margins of all plots
aligned_widths <- align_margin(map(l_grobs, \(x) {x$widths}), 'first')

# Setting the dimensions of plots to the aligned dimensions
for(i in seq_along(l_grobs)) l_grobs[[i]]$widths <- aligned_widths[[i]]

# Draw aligned plots
plot_grid(plotlist = l_grobs, nrow = 1, labels = 'AUTO')
```


A
Resource abundance, E(*R*)
0.85
0.80
0.75
0.70
0.65
2017−07
2018−01
2018−07
B
Resource stochasticity, Var(*R*)
0.005
0.004
0.003
0.002
0.001
2017−07
2018−01
2018−07
C
7−day home range size (km^2)
12.5
10.0
7.5
5.0
2.5
2017−07
2018−01
2018−07

To create the right side of the figure, we will need to estimate the effects of E(R) and Var(R) on the tapir's space use. To do this, we fit a GAM to the the tapir's 7-day home range estimates using the mean and variance in NDVI as predictors. However, since the two predictors are highly correlated, using a regular GAM causes the effects to be non-monotonic (after strongly restricting `k` to avoid over-fitting; see fig. C5).

We can now fit a GAM similar to the one used for the simulated data. However, since this dataset is much smaller and autocorrelated, we should be more parsimonious with the number of terms and their complexity. A model with a Gamma family of distributions and relatively smooth (i.e., not wiggly) marginal terms of $\mu(t)$ and $\sigma^2(t)$ is a reasonable start. This dataset is too small (and autocorrelated) to estimate changes in scale parameters with reasonable accuracy and precision.

```r
gam(hr_est_95 ~ s(mu, bs = 'ts', k = 5) + s(sigma2, bs = 'ts', k = 5),
    family = Gamma('log'),
    weights = weight, # weights based on CI size
    data = tapir,
    method = 'REML') %>%
  plot(pages = 1, scheme = 1) # trends should be monotonic
```

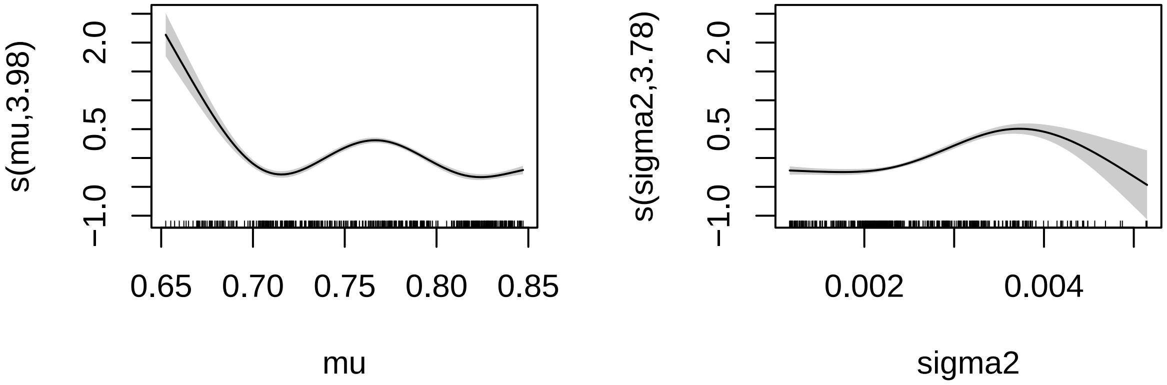


Figure C5: Marginal effects of $\mu(t)$ and $\sigma^2(t)$ on the tapir's space use (on the log link scale) using a GAM with no shape constraints. The estimated degrees of freedom for each term can be seen in parentheses in the y-axis labels. Shaded areas inticate the 95% credible intervals.

Due to the small sample size, the spatiotemporal autocorrelation, and the correlation between $\mu(t)$ and $\sigma^2(t)$, it is hard to separate the effects of the two predictors. However, since it is reasonable to expect the effects of $\mu(t)$ and $\sigma^2(t)$ to be monotonic decreaseing and increasing, respectively, we can use a Shape-Constrained Additive Model (SCAM) via the `scam` package (Pya 2022).

```r
m <- scam(hr_est_95 ~
            s(mu, bs = 'mpd', k = 5) +
            s(sigma2, bs = 'mpi', k = 5),
          family = Gamma('log'),
          weights = weight,
          data = tapir)
plot(m, pages = 1, scheme = 1)
```

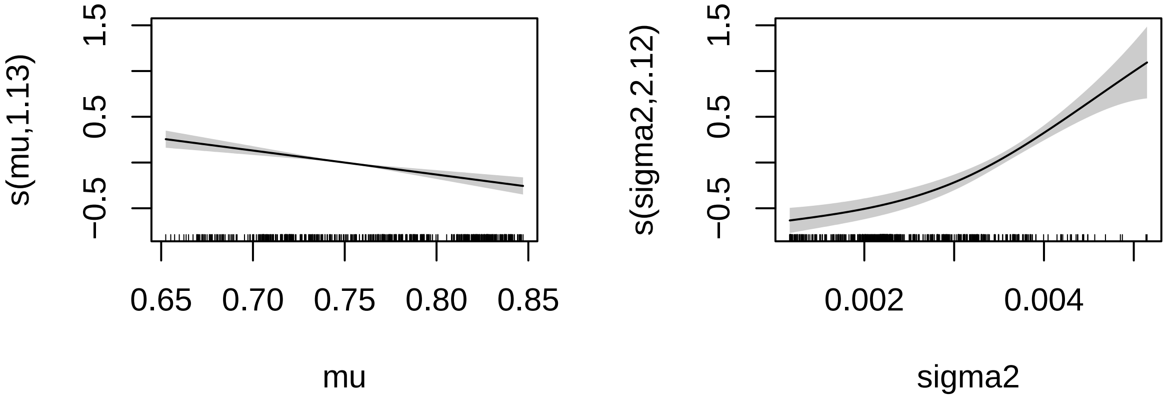


Figure C6: Marginal effects of $\mu(t)$ and $\sigma^2(t)$ on the tapir's space use (on the log link scale) using a shape-constrained GAM. The estimated degrees of freedom for each term can be seen in parentheses in the y-axis labels. Shaded areas inticate the 95% credible intervals.

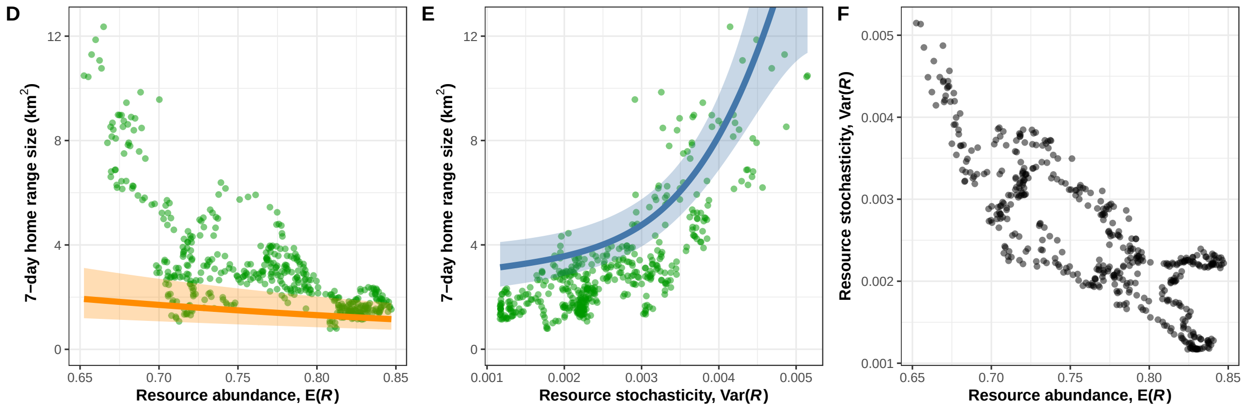
This SCAM fits substantially better than a SCAM that only accounts for $\mu(t)$:

```r
m_mu <- scam(hr_est_95 ~ s(mu, bs = 'mpd', k = 5),
             family = Gamma('log'),
             weights = weight,
             data = tapir)
round(AIC(m, m_mu), 3)
```

```
        df    AIC
m    5.232 907.57
m_mu 3.000 995.03
```

We can now create the predictions from the shape-constrained GAM.

```
theme_set(theme_bw()) # need to set the theme again for some reason
# predict marginal effects for each term
preds <- tibble(mu = gratia:::seq_min_max(tapir$mu, n = 250),
                sigma2 = gratia:::seq_min_max(tapir$sigma2, n = 250)) %>%
  bind_cols(
    # predictions for the marginal effect of mu
    predict(m, newdata = ., terms = 's(mu)', type = 'link',
            se.fit = TRUE) %>%
      as.data.frame() %>% # convert list to data frame
      transmute(hr_mu_est = exp(fit), # transform to the response scale
                hr_mu_lwr = exp(fit - 1.96 * se.fit), # 95% cred. intervals
                hr_mu_upr = exp(fit + 1.96 * se.fit)),
    # predictions for the marginal effect of sigma2
    predict(m, newdata = ., terms = 's(sigma2)', type = 'link',
            se.fit = TRUE) %>%
      as.data.frame() %>%
      transmute(hr_sigma2_est = exp(fit),
                hr_sigma2_lwr = exp(fit - 1.96 * se.fit),
                hr_sigma2_upr = exp(fit + 1.96 * se.fit)))

p_d <- ggplot() +
  coord_cartesian(ylim = c(0, 12.5)) +
  geom_point(aes(mu, hr_est_95), tapir, color = pal[3], alpha = 0.5) +
  geom_ribbon(aes(mu, ymin = hr_mu_lwr, ymax = hr_mu_upr), preds,
              fill = pal[1], alpha = 0.3) +
  geom_line(aes(mu, hr_mu_est), preds, color = pal[1], linewidth = 2) +
  xlab(e_r) +
  scale_y_continuous(hr_lab)

p_e <- ggplot() +
  coord_cartesian(ylim = c(0, 12.5)) +
  geom_point(aes(sigma2, hr_est_95), tapir, color = pal[3], alpha = 0.5) +
  geom_ribbon(aes(sigma2, ymin = hr_sigma2_lwr, ymax = hr_sigma2_upr),
              preds, fill = pal[2], alpha = 0.3) +
  geom_line(aes(sigma2, hr_sigma2_est), preds, color = pal[2],
            linewidth = 2) +
  xlab(v_r) +
  scale_y_continuous(hr_lab)

p_f <- ggplot(tapir) +
  geom_point(aes(mu, sigma2), alpha = 0.5) +
  labs(x = e_r, y = v_r)

r_grobs <- map(list(p_d, p_e, p_f), as_grob)

# align right margins of all plots
aligned_widths <- align_margin(map(r_grobs, \(x) {x$widths}), 'first')

# Setting the dimensions of plots to the aligned dimensions
for(i in seq_along(r_grobs)) r_grobs[[i]]$widths <- aligned_widths[[i]]

plot_grid(plotlist = r_grobs, ncol = 3, labels = c('D', 'E', 'F'))
```



References

Busetto, L., and L. Ranghetti. 2016. MODIStsp : An R package for automatic preprocessing of MODIS Land Products time series. Computers & Geosciences 97:40–48.

Fleming, C. H., and J. M. Calabrese. 2021. Ctmm: Continuous-Time Movement Modeling.

Grolemund, G., and H. Wickham. 2011. Dates and Times Made Easy with lubridate. Journal of Statistical Software 40:1–25.

Henry, L., and H. Wickham. 2022. Purrr: Functional Programming Tools.

Hijmans, R. J. 2023. Terra: Spatial Data Analysis.

Medici, E. P., S. Mezzini, C. H. Fleming, J. M. Calabrese, and M. J. Noonan. 2022. Movement ecology of vulnerable lowland tapirs between areas of varying human disturbance. Movement Ecology 10:14.

Noonan, M. J., M. A. Tucker, C. H. Fleming, T. S. Akre, S. C. Alberts, A. H. Ali, J. Altmann, et al. 2019. A comprehensive analysis of autocorrelation and bias in home range estimation. Ecological Monographs 89:e01344.

Pebesma, E. 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10:439.

Pettorelli, N., S. Ryan, T. Mueller, N. Bunnefeld, B. Jedrzejewska, M. Lima, and K. Kausrud. 2011. The Normalized Difference Vegetation Index (NDVI): Unforeseen successes in animal ecology. Climate Research 46:15–27.

Pettorelli, N., J. O. Vik, A. Mysterud, J.-M. Gaillard, C. J. Tucker, and N. Chr. Stenseth. 2005. Using the satellite-derived NDVI to assess ecological responses to environmental change. Trends in Ecology & Evolution 20:503–510.

Pya, N. 2022. Scam: Shape Constrained Additive Models.

Silva, I., C. H. Fleming, M. J. Noonan, J. Alston, C. Folta, W. F. Fagan, and J. M. Calabrese. 2022. Autocorrelation-informed home range estimation: A review and practical guide. Methods in Ecology and Evolution 13:534–544.

Stasinopoulos, M. D., and R. A. Rigby. 2007. Generalized additive models for location scale and shape (GAMLSS) in R. Journal of Statistical Software 23.

Wickham, H. 2016. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

Wickham, H., R. François, L. Henry, and K. Müller. 2022. Dplyr: A Grammar of Data Manipulation.

Wickham, H., and M. Girlich. 2022. Tidyr: Tidy Messy Data.

Wilke, C. O. 2020. Cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'.

Wood, S. N. 2017. Generalized additive models: An introduction with R. Chapman & Hall/CRC texts in statistical science (Second edition.). CRC Press/Taylor & Francis Group, Boca Raton.