

Appendix S2 - R Scripts for reproducing the results presented in the main text

EP Medici, S Mezzini, CH Fleming, JM Calabrese, MJ Noonan

11/01/2022

In this appendix we provide details on the analyses and R code used to generate all of the results presented in the main text. For full details on these analyses see the github repository associated with this manuscript at: <https://github.com/StefanoMezzini/tapirs>.

Pre-analysis setup

```
library('dplyr')      # for data wrangling
library('purrr')      # for functional programming
library('mgcv')       # for fitting GAMs
library('ctmm')       # for animal movement analyses
library('tidyr')      # for data wrangling
library('MuMIn')      # for model selection
library('sf')         # to work with spData maps
library('sp')         # to import spatial layers
library('raster')     # to import human footprint index raster
library('ggplot2')    # for plotting
library('cowplot')    # for plot grids
library('gratia')     # for seq_min_max(), draw()
theme_set(theme_bw())

# color palette
pal <- c('#4477AA', '#ff8c00', '#66CCEE', '#009900',
         '#CCBB44', '#EE6677', '#AA3377', '#BBBBBB')

N <- 74 # number of tapirs

tapirs <- readRDS('../models/tapirs-final.rds') %>% # tapir data
mutate(
  # convert tau_p from seconds to days
  tau.position.est = tau.position.est / (60^2 * 24),
  tau.position.low = tau.position.low / (60^2 * 24),
  tau.position.high = tau.position.high / (60^2 * 24),
  # convert tau_v from seconds to hours
  tau.velocity.est = tau.velocity.est / (60^2),
  tau.velocity.low = tau.velocity.low / (60^2),
  tau.velocity.high = tau.velocity.high / (60^2))
```

Individual variation in movement and space use

Mean home range estimates

```
meta(tapirs$model, plot = FALSE) %>% round(2)
```

```
##                                <U+0394>AICc
## inverse-Gaussian              0.00
## Dirac-d                       17382.45

##          low  est  high
## mean (km²) 6.53 8.31 10.42
## CoV² (RVAR) 0.65 1.02  1.48
## CoV (RSTD) 0.81 1.02  1.22
```

The overall mean home range estimate was 8.31 km² with CI (6.53, 10.4) km².

```
range(tapirs$area.est) %>% round(digits = 1)
```

```
## [1]  1.0 29.7
```

The range in home range area estimates was 1.0 - 29.7 km².

```
est <- function(y, parameter = c('mean', 'lwr', 'upr')) {
  if(sum(!is.na(y)) > 2) { # if there's at least 2 observations
    # fit a Gamma GLM with only an intercept to estimate the group-level mean
    m <- gam(y ~ 1,
             family = Gamma(link = 'log'),
             method = 'REML')

    if(length(parameter) > 1) stop('Specify a single parameter.\n')

    pred <- predict(m, newdata = tibble(c = 1), se.fit = TRUE, scale = 'link') %>%
      as.data.frame() %>%
      mutate(lwr = exp(fit - 1.96 * se.fit),
             mean = exp(fit),
             upr = exp(fit + 1.96 * se.fit))
    pred[1, parameter]
  } else {NA_real_}
}

tap <-
  bind_rows(
    bind_cols(
      name = c('Atlantic forest', 'Pantanal', 'Cerrado', 'Overall'),
      region.lab = c('Atlantic forest', 'Pantanal', 'Cerrado', 'Overall'),
      bind_rows(
        meta(tapirs$model[tapirs$region == 'atlantica'], plot = FALSE)[1, ],
        meta(tapirs$model[tapirs$region == 'pantanal'], plot = FALSE)[1, ],
        meta(tapirs$model[tapirs$region == 'cerrado'], plot = FALSE)[1, ],
        meta(tapirs$model, plot = FALSE)[1, ]) %>%
        rename(area.low = low, area.est = est, area.high = high),
      # regional estimates
      group_by(tapirs, region) %>%
        summarize(tau.position.e = est(tau.position.est, 'mean'),
                  tau.position.low = est(tau.position.est, 'lwr'),
                  tau.position.high = est(tau.position.est, 'upr'),
```

```

    tau.velocity.e = est(tau.velocity.est, 'mean'),
    tau.velocity.low = est(tau.velocity.est, 'lwr'),
    tau.velocity.high = est(tau.velocity.est, 'upr'),
    speed.e = est(speed.est, 'mean'),
    speed.low = est(speed.est, 'lwr'),
    speed.high = est(speed.est, 'upr')) %>%
  rename(tau.position.est = tau.position.e,
    tau.velocity.est = tau.velocity.e,
    speed.est = speed.e) %>%
  bind_rows(
    tapirs %>%
      summarize(tau.position.est = est(tapirs$tau.position.est, 'mean'),
        tau.position.low = est(tapirs$tau.position.est, 'lwr'),
        tau.position.high = est(tapirs$tau.position.est, 'upr'),
        tau.velocity.est = est(tapirs$tau.velocity.est, 'mean'),
        tau.velocity.low = est(tapirs$tau.velocity.est, 'lwr'),
        tau.velocity.high = est(tapirs$tau.velocity.est, 'upr'),
        speed.est = est(tapirs$speed.est, 'mean'),
        speed.low = est(tapirs$speed.est, 'lwr'),
        speed.high = est(tapirs$speed.est, 'upr')))) %>%
  dplyr::select(region.lab, area.est, area.low, area.high, tau.position.est,
    tau.position.low, tau.position.high, tau.velocity.est,
    tau.velocity.low, tau.velocity.high, speed.est, speed.low,
    speed.high)

```

```

##                               <U+0394>AICc
## inverse-Gaussian             0.000
## Dirac-d                      1310.217
##                               <U+0394>AICc
## inverse-Gaussian             0.000
## Dirac-d                      8562.481
##                               <U+0394>AICc
## inverse-Gaussian             0.000
## Dirac-d                      6961.898
##                               <U+0394>AICc
## inverse-Gaussian             0.00
## Dirac-d                      17382.45
knitr::kable(t(tap)) # transpose to make readable

```

region.lab	Atlantic forest	Pantanal	Cerrado	Overall
area.est	10.285217	5.936260	13.383010	8.307126
area.low	4.381964	4.695495	7.652765	6.527059
area.high	20.741634	7.400505	21.797593	10.418793
tau.position.est	0.5393670	0.6555289	0.8000626	0.7227463
tau.position.low	0.1883971	0.3981069	0.3633637	0.4191364
tau.position.high	1.544168	1.079404	1.761596	1.246282
tau.velocity.est	NA	0.4004825	0.4610756	0.4388269
tau.velocity.low	NA	0.3504944	0.3732013	0.3757900
tau.velocity.high	NA	0.4576000	0.5696408	0.5124378
speed.est	NA	12.14095	10.67056	11.17040
speed.low	NA	10.638837	9.325726	10.107009
speed.high	NA	13.85514	12.20933	12.34567

```
round(range(tapirs$tau.position.est, na.rm = TRUE) / 60 / 60 / 24, 2)
```

```
## [1] 0 0
```

The range in range crossing time was 0.05-12.8 days.

```
round(range(tapirs$tau.velocity.est, na.rm = TRUE) / 60 / 60 , 2)
```

```
## [1] 0 0
```

The range in directional persistence timescale was 0.17-1.88 hours.

Estimate differences in mean daily movement speed between animals of different ages and sexes using Gamma GLMs fit via mgcv: `gam()`:

```
# Differences in movement speed between animals of different ages and sexes
```

```
m.speed.sex <- gam(speed.est ~ sex,  
                  family = Gamma('log'),  
                  data = tapirs,  
                  method = 'REML')
```

```
m.speed.age <- gam(speed.est ~ adult,  
                  family = Gamma('log'),  
                  data = tapirs,  
                  method = 'REML')
```

```
# model summaries
```

```
summary(m.speed.sex)
```

```
##
```

```
## Family: Gamma
```

```
## Link function: log
```

```
##
```

```
## Formula:
```

```
## speed.est ~ sex
```

```
##
```

```
## Parametric coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  2.35276    0.06872  34.235  <2e-16 ***  
## sexMALE      0.12420    0.10006   1.241    0.22
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
##
```

```
## R-sq.(adj) =  0.00946   Deviance explained = 2.16%
```

```
## -REML = 156.04   Scale est. = 0.13224    n = 53
```

```
summary(m.speed.age)
```

```
##
```

```
## Family: Gamma
```

```
## Link function: log
```

```
##
```

```
## Formula:
```

```
## speed.est ~ adult
```

```
##
```

```
## Parametric coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  2.2532    0.0928  24.281  <2e-16 ***
```

```
## adultYes      0.2168      0.1096      1.978      0.0533 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.0454   Deviance explained = 5.18%
## -REML = 155.11   Scale est. = 0.12917   n = 53

# predictions
tibble(Sex = c('MALE', 'FEMALE'),
       bind_cols(predict(m.speed.sex, tibble(sex = Sex), se.fit = TRUE))) %>%
  mutate(est = round(exp(fit), 2),
         lwr = round(exp(fit - 1.96 * se.fit), 2),
         upr = round(exp(fit + 1.96 * se.fit), 2))

## # A tibble: 2 x 6
##   Sex      fit se.fit  est  lwr  upr
##   <chr>  <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 MALE    2.48 0.0727  11.9 10.3  13.7
## 2 FEMALE  2.35 0.0687  10.5  9.19 12.0

tibble(Adult = c('Yes', 'No'),
       bind_cols(predict(m.speed.age, tibble(adult = Adult), se.fit = TRUE))) %>%
  mutate(est = round(exp(fit), 2),
         lwr = round(exp(fit - 1.96 * se.fit), 2),
         upr = round(exp(fit + 1.96 * se.fit), 2))

## # A tibble: 2 x 6
##   Adult  fit se.fit  est  lwr  upr
##   <chr>  <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Yes    2.47 0.0583  11.8 10.6  13.2
## 2 No    2.25 0.0928  9.52  7.94 11.4

Estimate differences in home range sizes using ctm::meta():

# by sex
meta(list(female = filter(tapirs, sex == 'FEMALE')$akde,
         male = filter(tapirs, sex == 'MALE')$akde),
      plot = FALSE)

## * Sub-population female
##
##           <U+0394>AICc
## inverse-Gaussian      0.000
## Dirac-d              7367.412
##
## * Sub-population male
##
##           <U+0394>AICc
## inverse-Gaussian      0.000
## Dirac-d              7240.514
##
## * Joint population
##
##           <U+0394>AICc
## inverse-Gaussian      0.00
## Dirac-d              14778.71
##
## * Joint population versus sub-populations (best models)
```

```
##                                <U+0394>AICc
## Joint population              0.000000
## Sub-population                4.035555

## , , low
##
##                /female      /male
## female/ 1.0000000 0.7026961
## male/    0.5608837 1.0000000
##
## , , est
##
##                /female      /male
## female/ 1.000000 1.094326
## male/    0.873601 1.000000
##
## , , high
##
##                /female      /male
## female/ 1.000000 1.629906
## male/    1.300086 1.000000

meta(filter(tapirs, sex == 'FEMALE')$akde, plot = FALSE) # CIs for females
```

```
##                                <U+0394>AICc
## inverse-Gaussian              0.000
## Dirac-d                      7367.412

##                low      est      high
## mean (km2) 4.5262480 6.1131626 8.068986
## CoV2 (RVAR) 0.4615141 0.8615983 1.384492
## CoV (RSTD) 0.6857438 0.9369616 1.187722

meta(filter(tapirs, sex == 'MALE')$akde, plot = FALSE) # CIs for males
```

```
##                                <U+0394>AICc
## inverse-Gaussian              0.000
## Dirac-d                      7240.514

##                low      est      high
## mean (km2) 4.0316213 5.4605383 7.232389
## CoV2 (RVAR) 0.3590258 0.7091027 1.176316
## CoV (RSTD) 0.6057261 0.8512715 1.096416

# by age group
meta(list(subadult = filter(tapirs, adult == 'No')$akde,
      adult = filter(tapirs, adult == 'Yes')$akde),
      plot = FALSE)
```

```
## * Sub-population subadult

##                                <U+0394>AICc
## inverse-Gaussian              0.000
## Dirac-d                      4785.763

## * Sub-population adult

##                                <U+0394>AICc
## inverse-Gaussian              0.000
```

```

## Dirac-d                9841.229
## * Joint population
##                        <U+0394>AICc
## inverse-Gaussian       0.00
## Dirac-d                14778.71
## * Joint population versus sub-populations (best models)
##                        <U+0394>AICc
## Sub-population         0.00000
## Joint population       3.08865
## , , low
##
##           /subadult   /adult
## subadult/ 1.0000000 0.6457536
## adult/    0.3654864 1.0000000
##
## , , est
##
##           /subadult   /adult
## subadult/ 1.0000000 1.287812
## adult/    0.7053557 1.000000
##
## , , high
##
##           /subadult   /adult
## subadult/ 1.000000 2.179042
## adult/    1.321561 1.000000
meta(filter(tapirs, adult == 'No')$akde, plot = FALSE) # CIs for sub-adults

##                        <U+0394>AICc
## inverse-Gaussian       0.000
## Dirac-d                4785.763
##
##           low      est      high
## mean (km2) 3.8693250 6.985178 11.648652
## CoV2 (RVAR) 0.5763867 1.674818 3.348694
## CoV (RSTD) 0.7764644 1.323575 1.871554
meta(filter(tapirs, adult == 'Yes')$akde, plot = FALSE) # CIs for adults

##                        <U+0394>AICc
## inverse-Gaussian       0.000
## Dirac-d                9841.229
##
##           low      est      high
## mean (km2) 4.3936502 5.3698352 6.4954695
## CoV2 (RVAR) 0.3073039 0.5063385 0.7542638
## CoV (RSTD) 0.5578841 0.7161111 0.8740204

```

Variation in movement across biomes and habitat composition (figure 5)

The best model is found using `MuIn::dredge()`. Smooth terms are allowed to have a maximum `k` of 10 (or 5 if there are not enough unique values). The models are fit using maximum likelihood while using `MuIn::dredge()`, but the best model is fit using Restricted Maximum Likelihood (REML).

```
tapirs.lu <- readRDS('../models/tapirs-land-use.rds') %>%
  mutate(tau.velocity.est = tau.velocity.est / (60^2), # from seconds to hours
         tau.velocity.low = tau.velocity.low / (60^2),
         tau.velocity.high = tau.velocity.high / (60^2))
tapirs.lu.sp <- filter(tapirs.lu, !is.na(speed.est))
tapirs.lu.tv <- filter(tapirs.lu, !is.na(tau.velocity.est))

# only use groups with at least 5 unique values
dplyr::select(tapirs.lu, `?`:plantation) %>%
  pivot_longer(-c()) %>%
  group_by(name) %>%
  summarize(unique = length(unique(value)),
           min = min(value),
           max = max(value)) %>%
  arrange(desc(unique)) %>%
  mutate(`>5` = unique > 5,
         `>10` = unique > 10)
```

```
## # A tibble: 11 x 6
##   name          unique   min    max `>5` `>10`
##   <chr>          <int> <dbl> <dbl> <lgl> <lgl>
## 1 forest           69 0.139 1      TRUE  TRUE
## 2 floodplain       62 0      0.656 TRUE  TRUE
## 3 savannah        18 0      0.611 TRUE  TRUE
## 4 dirt            12 0      0.221 TRUE  TRUE
## 5 pasture         11 0      0.352 TRUE  TRUE
## 6 crop              7 0      0.412 TRUE FALSE
## 7 water            7 0      0.0942 TRUE FALSE
## 8 ?                6 0      0.181 TRUE FALSE
## 9 headquarters    6 0      0.0178 TRUE FALSE
## 10 plantation      3 0      0.324 FALSE FALSE
## 11 urban           2 0      0.0210 FALSE FALSE
```

```
# Gamma GAM regression on home range estimate
dredge(global.model = gam(area.est ~
  s(forest, k = 10) +
  s(floodplain, k = 10) +
  s(savannah, k = 10) +
  s(dirt, k = 10) +
  s(pasture, k = 5) +
  s(crop, k = 5) +
  s(water, k = 5),
  family = Gamma('log'),
  data = tapirs.lu,
  na.action = na.fail,
  method = 'ML'), # need to use ML with dredge()

  rank = 'AICc') %>%
head(20)
```



```

## Fixed term is "(Intercept)"

## Global model call: gam(formula = area.est ~ s(forest, k = 10) + s(floodplain, k = 10) +
##      s(savannah, k = 10) + s(dirt, k = 10) + s(pasture, k = 5) +
##      s(crop, k = 5) + s(water, k = 5), family = Gamma("log"),
##      data = tapirs.lu, na.action = na.fail, method = "ML")
## ---
## Model selection table
##      (Int) s(crp,5) s(drt,10) s(fld,10) s(frs,10) s(pst,5) s(svn,10) s(wtr,5) df
## 11  1.649                +                +                8
## 75  1.644                +                +                + 9
## 7   1.656                +                +                8
## 79  1.635                +                +                + 11
## 15  1.642                +                +                10
## 71  1.650                +                +                + 9
## 43  1.645                +                +                + 10
## 107 1.639                +                +                + 11
## 12  1.650                +                +                9
## 76  1.643                +                +                + 10
## 27  1.649                +                +                9
## 23  1.655                +                +                9
## 8   1.657                +                +                8
## 80  1.635                +                +                + 11
## 91  1.644                +                +                + 10
## 16  1.642                +                +                10
## 39  1.658                +                +                + 8
## 108 1.637                +                +                + 11
## 3   1.685                +                +                5
## 44  1.645                +                +                + 10
##      logLik  AICc delta weight
## 11  -181.641 383.7  0.00  0.126
## 75  -180.574 384.0  0.29  0.109
## 7   -182.865 384.2  0.51  0.098
## 79  -178.892 384.9  1.17  0.070
## 15  -180.273 384.9  1.18  0.070
## 71  -181.781 385.4  1.73  0.053
## 43  -180.854 385.4  1.74  0.053
## 107 -179.564 385.6  1.86  0.050
## 12  -181.662 385.6  1.92  0.048
## 76  -180.493 385.7  2.02  0.046
## 27  -181.611 386.3  2.61  0.034
## 23  -182.612 386.4  2.73  0.032
## 8   -182.964 386.5  2.78  0.031
## 80  -178.829 386.6  2.87  0.030
## 91  -180.526 386.6  2.94  0.029
## 16  -180.298 386.7  3.05  0.027
## 39  -183.218 386.8  3.14  0.026
## 108 -179.276 387.1  3.39  0.023
## 3   -187.903 387.2  3.48  0.022
## 44  -180.835 387.3  3.65  0.020
## Models ranked by AICc(x)

```

Model 3 with `s(dirt)` as the only predictor is the best (parsimonious) model, since it is the best single-term model, and its AICc is not much larger than any of the models with a lower AICc (`delta < 3.5`). Still, the effect of `s(dirt)` is negligible.

```

m.hr <- gam(area.est ~ s(dirt, k = 5),
             family = Gamma('log'),
             data = tapirs.lu,
             method = 'REML')
summary(m.hr)

##
## Family: Gamma
## Link function: log
##
## Formula:
## area.est ~ s(dirt, k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.68094    0.09319   18.04  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df    F p-value
## s(dirt) 2.987  3.493 3.495  0.0228 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.477  Deviance explained = 28.6%
## -REML = 193.89  Scale est. = 0.64264  n = 74
# Gamma GAM regression on average speed estimate
MuMIn::dredge(global.model = gam(speed.est ~
                                s(forest, k = 10) +
                                s(floodplain, k = 10) +
                                s(savannah, k = 10) +
                                s(dirt, k = 10) +
                                s(pasture, k = 5) +
                                s(crop, k = 5) +
                                s(water, k = 5),
                                family = Gamma('log'),
                                data = tapirs.lu.sp, # cannot have NAs
                                na.action = na.fail,
                                method = 'ML'),
                                rank = 'AICc') %>%
head(20)

## Fixed term is "(Intercept)"
## Global model call: gam(formula = speed.est ~ s(forest, k = 10) + s(floodplain, k = 10) +
## s(savannah, k = 10) + s(dirt, k = 10) + s(pasture, k = 5) +
## s(crop, k = 5) + s(water, k = 5), family = Gamma("log"),
## data = tapirs.lu.sp, na.action = na.fail, method = "ML")
## ---
## Model selection table
##      (Int) s(crp,5) s(drt,10) s(fld,10) s(frs,10) s(pst,5) s(svn,10) s(wtr,5) df
## 73  2.400                                +                                + 4
## 105 2.399                                +                                + 5

```

```

## 77 2.399          +          +          + 5
## 75 2.399          +          +          + 5
## 74 2.400          +          +          + 5
## 89 2.400          +          +          + 5
## 97 2.405          +          +          + 4
## 9  2.409          +          +          + 3
## 65 2.409          +          +          + 3
## 107 2.399          +          +          + 6
## 106 2.399          +          +          + 6
## 101 2.402          +          +          + 5
## 33 2.410          +          +          + 3
## 1  2.413          +          +          + 2
## 109 2.399          +          +          + 6
## 121 2.399          +          +          + 6
## 93 2.399          +          +          + 6
## 79 2.399          +          +          + 6
## 78 2.399          +          +          + 6
## 103 2.399          +          +          + 6
##      logLik  AICc delta weight
## 73 -148.964 306.8  0.00  0.226
## 105 -148.673 308.6  1.86  0.089
## 77 -148.703 308.7  1.92  0.086
## 75 -148.844 309.0  2.20  0.075
## 74 -148.919 309.1  2.35  0.070
## 89 -148.946 309.2  2.41  0.068
## 97 -150.688 310.2  3.45  0.040
## 9  -152.014 310.5  3.76  0.035
## 65 -152.130 310.7  3.99  0.031
## 107 -148.568 311.0  4.20  0.028
## 106 -148.604 311.0  4.27  0.027
## 101 -149.882 311.0  4.28  0.027
## 33 -152.289 311.1  4.31  0.026
## 1  -153.438 311.1  4.35  0.026
## 109 -148.663 311.2  4.39  0.025
## 121 -148.672 311.2  4.41  0.025
## 93 -148.682 311.2  4.43  0.025
## 79 -148.683 311.2  4.43  0.025
## 78 -148.697 311.2  4.46  0.024
## 103 -148.733 311.3  4.53  0.023
## Models ranked by AICc(x)

```

No term contributed significantly to the model, since no model had an AICc lower than 4.35 with respect to the null model with only an intercept term (`speed.est ~ 1`, see model 1 in the table above).

```

# Gamma GAM regression on average directional persistence (tau_v) estimate
filter(tapirs.lu, !is.na(tau.velocity.est)) %>%
  dplyr::select(`?`:plantation) %>%
  pivot_longer(-c()) %>%
  group_by(name) %>%
  summarize(unique = length(unique(value)),
            min = min(value),
            max = max(value)) %>%
  arrange(desc(unique)) %>%
  mutate(`>5` = unique > 5,
         `>10` = unique > 10) %>%

```

```

filter(`>5`)

## # A tibble: 6 x 6
##   name      unique  min    max `>5` `>10`
##   <chr>      <int> <dbl> <dbl> <lgl> <lgl>
## 1 forest      46 0.229 1      TRUE TRUE
## 2 floodplain  44 0      0.564 TRUE TRUE
## 3 savannah   15 0      0.611 TRUE TRUE
## 4 pasture     10 0      0.352 TRUE FALSE
## 5 dirt         9 0      0.221 TRUE FALSE
## 6 water        6 0      0.0942 TRUE FALSE

dredge(global.model = gam(tau.velocity.est ~
  s(forest, k = 10) +
  s(floodplain, k = 10) +
  s(savannah, k = 10) +
  s(dirt, k = 5) +
  s(pasture, k = 5) +
  s(crop, k = 5) +
  s(water, k = 5),
  family = Gamma('log'),
  na.action = na.fail,
  data = tapirs.lu.tv,
  method = 'ML'),
  rank = 'AICc') %>%
head(30)

## Fixed term is "(Intercept)"

## Global model call: gam(formula = tau.velocity.est ~ s(forest, k = 10) + s(floodplain,
##   k = 10) + s(savannah, k = 10) + s(dirt, k = 5) + s(pasture,
##   k = 5) + s(crop, k = 5) + s(water, k = 5), family = Gamma("log"),
##   data = tapirs.lu.tv, na.action = na.fail, method = "ML")
## ---
## Model selection table
##      (Int) s(crp,5) s(drt,5) s(fld,10) s(frs,10) s(pst,5) s(svn,10) s(wtr,5)
## 77 -0.8414                +          +                  +
## 105 -0.8409                +                  +          +
## 73 -0.8370                +                  +          +
## 79 -0.8421                +          +          +          +
## 71 -0.8391                +          +                  +
## 69 -0.8360                +                  +          +
## 78 -0.8417          +          +          +                  +
## 109 -0.8415                +          +                  +          +
## 121 -0.8415                +          +          +          +
## 103 -0.8415                +          +                  +          +
## 93 -0.8414                +          +          +          +
## 101 -0.8381                +                  +          +
## 106 -0.8410          +                  +          +          +
## 107 -0.8409                +          +          +          +
## 70 -0.8378          +                  +                  +          +
## 102 -0.8404          +          +                  +          +
## 89 -0.8371                +          +          +          +
## 75 -0.8370                +          +                  +          +
## 74 -0.8370          +                  +          +          +

```

```

## 65 -0.8299
## 87 -0.8391
## 72 -0.8391
## 85 -0.8360
## 111 -0.8421
## 80 -0.8421
## 95 -0.8421
## 119 -0.8418
## 94 -0.8418
## 110 -0.8417
## 125 -0.8417
##      df logLik  AICc delta weight
## 77    5 27.880 -44.3  0.00  0.138
## 105    5 27.673 -43.8  0.41  0.112
## 73     4 25.949 -42.9  1.34  0.071
## 79     6 28.200 -42.2  2.01  0.050
## 71     5 26.845 -42.2  2.07  0.049
## 69     4 25.520 -42.1  2.20  0.046
## 78     6 28.030 -41.9  2.35  0.043
## 109    6 27.939 -41.7  2.54  0.039
## 121    6 27.916 -41.7  2.58  0.038
## 103    6 27.913 -41.7  2.59  0.038
## 93     6 27.903 -41.7  2.61  0.037
## 101    5 26.426 -41.4  2.91  0.032
## 106    6 27.679 -41.2  3.06  0.030
## 107    6 27.673 -41.2  3.07  0.030
## 70     5 26.284 -41.1  3.19  0.028
## 102    6 27.440 -40.7  3.53  0.024
## 89     5 25.997 -40.5  3.77  0.021
## 75     5 25.965 -40.4  3.83  0.020
## 74     5 25.949 -40.4  3.86  0.020
## 65     3 23.118 -39.7  4.60  0.014
## 87     6 26.868 -39.6  4.68  0.013
## 72     6 26.854 -39.6  4.71  0.013
## 85     5 25.521 -39.5  4.72  0.013
## 111    7 28.214 -39.5  4.78  0.013
## 80     7 28.207 -39.5  4.79  0.013
## 95     7 28.204 -39.5  4.80  0.013
## 119    7 28.087 -39.2  5.03  0.011
## 94     7 28.042 -39.1  5.12  0.011
## 110    7 28.031 -39.1  5.15  0.011
## 125    7 28.028 -39.1  5.15  0.011
## Models ranked by AICc(x)

```

Model 73 ($\text{tau.velocity.est} \sim \text{s(forest, } k = 10) + \text{s(water, } k = 5)$) is the best two-term model. While it is not substantially better than model 65 ($\text{tau.velocity.est} \sim \text{s(water, } k = 5)$), the forest smooth term was included because of the high range in proportion of forested habitat (0.138-1).

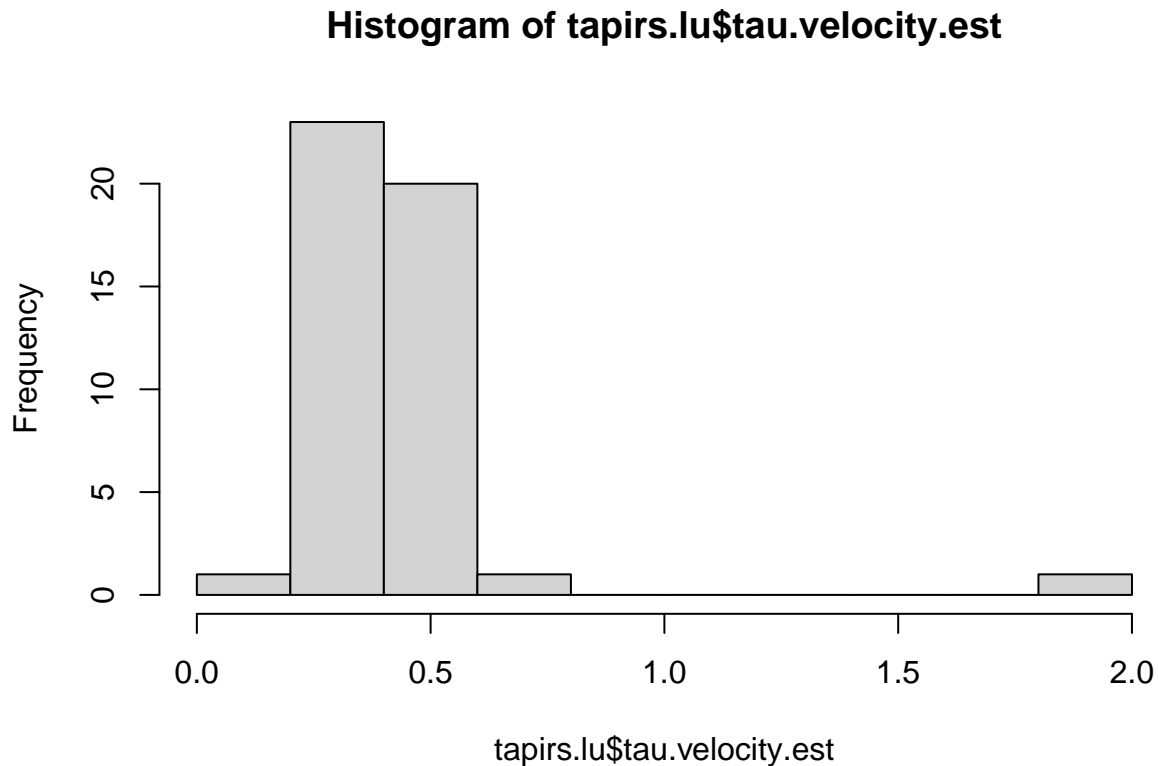
```

m.tauv <- gam(tau.velocity.est ~ s(forest, k = 10) + s(water, k = 5),
              family = Gamma('log'),
              data = tapirs.lu,
              method = 'REML')

```

We can re-fit the best models for τ_v without the outlier:

```
hist(tapirs.lu$tau.velocity.est)
```

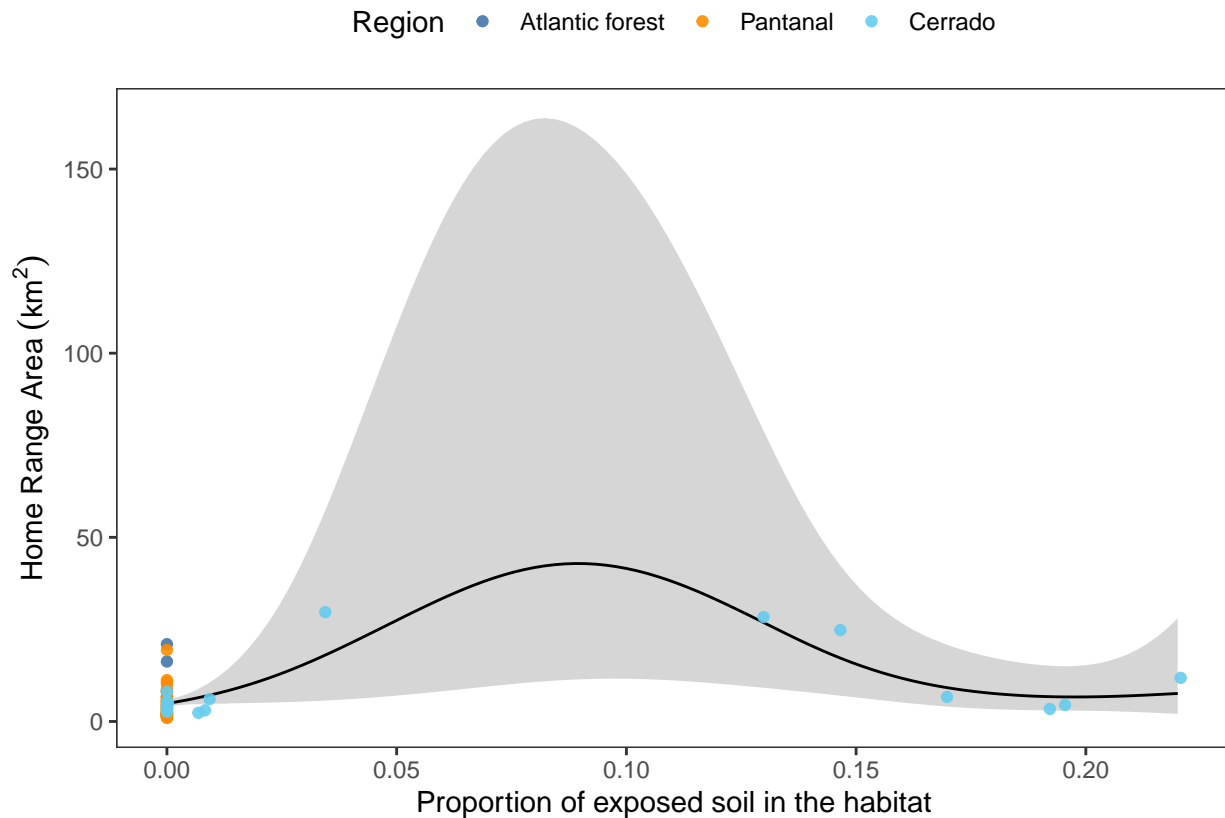


```
m.tauv.1 <- gam(tau.velocity.est ~ s(forest, k = 10) + s(water, k = 5),
  family = Gamma('log'),
  data = filter(tapirs.lu, tau.velocity.est < 1.5),
  method = 'REML')
```

Create predictions for each model and plot them:

```
# lu on hr (panel 5a)
pred.lu.hr <- tibble(dirt = seq(0, 0.22, length.out = 100))
pred.lu.hr <- bind_cols(pred.lu.hr,
  predict(m.hr, newdata = pred.lu.hr, se.fit = TRUE)) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

ggplot() +
  geom_ribbon(aes(dirt, ymin = lwr, ymax = upr), pred.lu.hr, alpha = 0.2) +
  geom_line(aes(dirt, est), pred.lu.hr) +
  geom_point(aes(dirt, area.est, color = region.lab), tapirs.lu, alpha = 0.9) +
  scale_color_manual('Region', values = pal[1:3],
    labels = c('Atlantic forest', 'Pantanal', 'Cerrado')) +
  labs(x = 'Proportion of exposed soil in the habitat',
    y = expression('Home Range Area'~(km^2))) +
  theme(legend.position = 'top',
    panel.grid = element_blank())
```



```
# lu on tau_v
pred.tv.f <- tibble(forest = seq_min_max(tapirs.lu$forest, n = 100),
  water = mean(tapirs.lu$water, na.rm = TRUE))
pred.tv.f <-
  bind_cols(pred.tv.f, predict(m.tauv, newdata = pred.tv.f, se.fit = TRUE,
    terms = 's(forest)')) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

pred.tv.w <- tibble(forest = mean(tapirs.lu$forest, na.rm = TRUE),
  water = seq_min_max(tapirs.lu$water, n = 100))
pred.tv.w <-
  bind_cols(pred.tv.w, predict(m.tauv, newdata = pred.tv.w, se.fit = TRUE,
    terms = 's(water)')) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

# regression plot
p.forest <-
  ggplot() +
  geom_ribbon(aes(forest, ymin = lwr, ymax = upr), pred.tv.f, alpha = 0.2) +
  geom_line(aes(forest, est), pred.tv.f) +
  geom_segment(aes(x = forest, xend = forest, y = tau.velocity.low,
    yend = tau.velocity.high, color = region.lab), tapirs.lu,
```

```

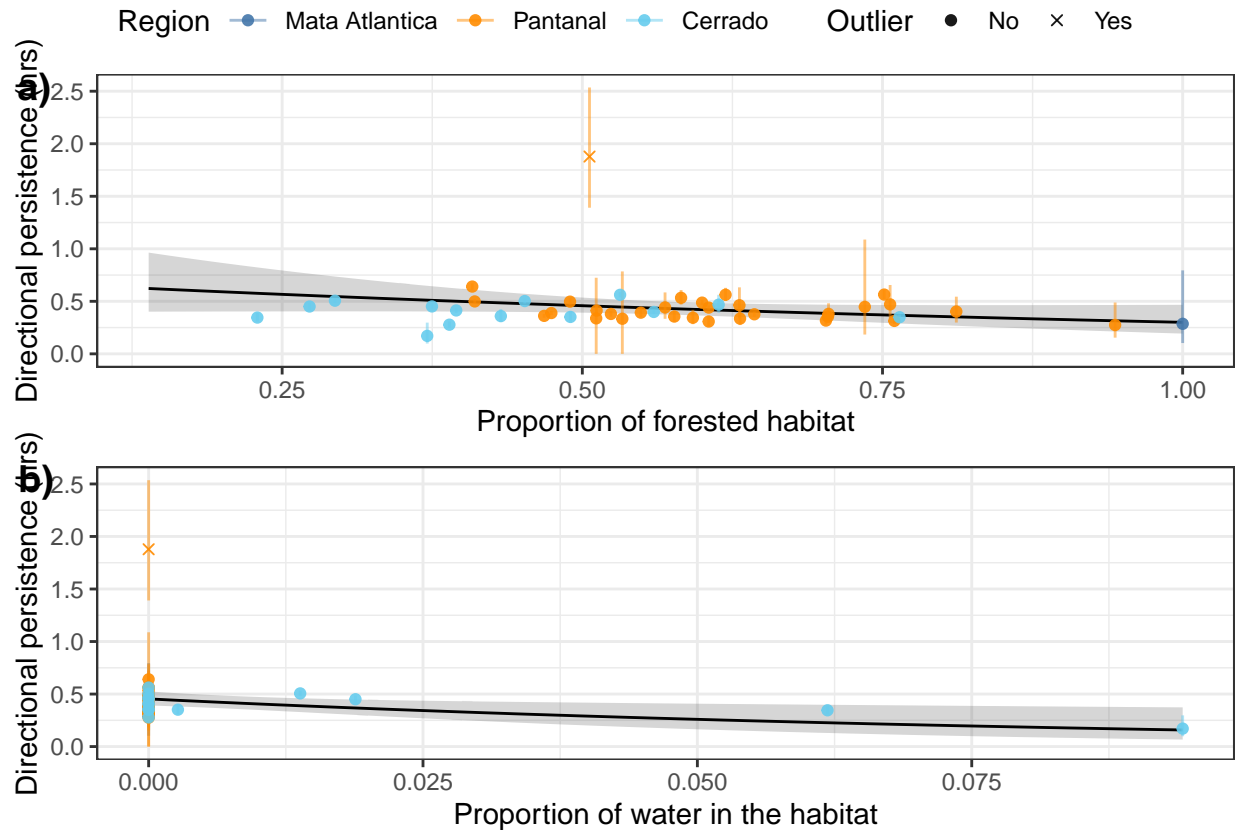
      lwd = 0.5, alpha = 0.5) +
geom_point(aes(forest, tau.velocity.est, color = region.lab,
               shape = tau.velocity.est > 1.5),
           filter(tapirs.lu, ! is.na(tau.velocity.est)), alpha = 0.9) +
scale_shape_manual('Outlier', values = c(19, 4), labels = c('No', 'Yes')) +
scale_color_manual('Region', values = pal[1:3]) +
labs(x = 'Proportion of forested habitat',
     y = 'Directional persistence (hrs)' ) +
theme(legend.position = 'none')

p.water <-
ggplot() +
geom_ribbon(aes(water, ymin = lwr, ymax = upr), pred.tv.w, alpha = 0.2) +
geom_line(aes(water, est), pred.tv.w) +
geom_segment(aes(x = water, xend = water, y = tau.velocity.low,
                 yend = tau.velocity.high, color = region.lab), tapirs.lu,
            lwd = 0.5, alpha = 0.5) +
geom_point(aes(water, tau.velocity.est, color = region.lab,
               shape = tau.velocity.est > 1.5),
           filter(tapirs.lu, ! is.na(tau.velocity.est)), alpha = 0.9) +
scale_shape_manual('Outlier', values = c(19, 4), labels = c('No', 'Yes')) +
scale_color_manual('Region', values = pal[1:3]) +
labs(x = 'Proportion of water in the habitat',
     y = 'Directional persistence (hrs)' ) +
theme(legend.position = 'none')

p <- plot_grid(get_legend(p.forest + theme(legend.position = 'top')),
               p.forest, p.water,
               ncol = 1, rel_heights = c(0.2, 1, 1),
               labels = c(NA, 'a', 'b')); p

## Warning: Removed 28 rows containing missing values (geom_segment).
## Warning: Removed 28 rows containing missing values (geom_segment).
## Warning: Removed 28 rows containing missing values (geom_segment).
## Warning: Removed 1 rows containing missing values (geom_text).

```

```
# Regression plot without outlier (panels 5b and 5c)
pred.tv.f <- tibble(forest = seq_min_max(tapirs.lu$forest, n = 100),
  water = mean(tapirs.lu$water, na.rm = TRUE))
pred.tv.f <-
  bind_cols(pred.tv.f, predict(m.tauv.1, newdata = pred.tv.f, se.fit = TRUE,
    terms = 's(forest)')) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

pred.tv.w <- tibble(forest = mean(tapirs.lu$forest, na.rm = TRUE),
  water = seq_min_max(tapirs.lu$water, n = 100))
pred.tv.w <-
  bind_cols(pred.tv.w, predict(m.tauv.1, newdata = pred.tv.w, se.fit = TRUE,
    terms = 's(water)')) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

p.forest <-
  ggplot() +
  geom_ribbon(aes(forest, ymin = lwr, ymax = upr), pred.tv.f, alpha = 0.2) +
  geom_line(aes(forest, est), pred.tv.f) +
  geom_segment(aes(x = forest, xend = forest, y = tau.velocity.low,
    yend = tau.velocity.high, color = region.lab),
```

```

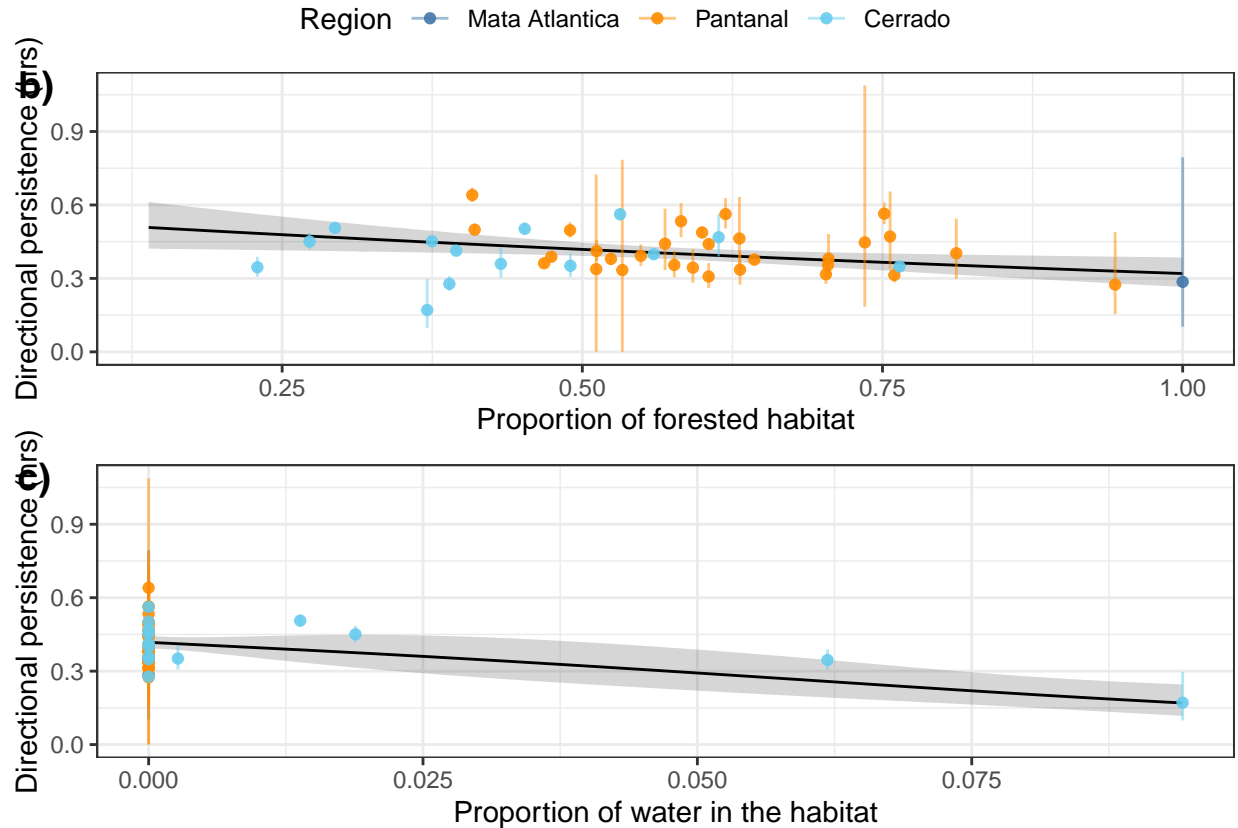
      filter(tapirs.lu, tau.velocity.est < 1.5), lwd = 0.5,
      alpha = 0.5) +
    geom_point(aes(forest, tau.velocity.est, color = region.lab),
      filter(tapirs.lu, tau.velocity.est < 1.5), alpha = 0.9) +
    scale_color_manual('Region', values = pal[1:3]) +
    labs(x = 'Proportion of forested habitat',
      y = 'Directional persistence (hrs)' ) +
    theme(legend.position = 'none')

p.water <-
  ggplot() +
    geom_ribbon(aes(water, ymin = lwr, ymax = upr), pred.tv.w, alpha = 0.2) +
    geom_line(aes(water, est), pred.tv.w) +
    geom_segment(aes(x = water, xend = water, y = tau.velocity.low,
      yend = tau.velocity.high, color = region.lab),
      filter(tapirs.lu, tau.velocity.est < 1.5),
      lwd = 0.5, alpha = 0.5) +
    geom_point(aes(water, tau.velocity.est, color = region.lab),
      filter(tapirs.lu, tau.velocity.est < 1.5), alpha = 0.9) +
    scale_color_manual('Region', values = pal[1:3]) +
    labs(x = 'Proportion of water in the habitat',
      y = 'Directional persistence (hrs)' ) +
    theme(legend.position = 'none')

plot_grid(get_legend(p.forest + theme(legend.position = 'top')),
  p.forest, p.water,
  ncol = 1, rel_heights = c(0.2, 1, 1, 1),
  labels = c(NA, 'b', 'c'))

## Warning: Removed 1 rows containing missing values (geom_text).

```



As requested by reviewer #2, we can repeat the analysis for panels 5b and 5c without the data from the Atlantic forest biome (after having removed the outlier):

```
m.tauv.2 <- gam(tau.velocity.est ~ s(forest, k = 10) + s(water, k = 5),
  family = Gamma('log'),
  data = filter(tapirs.lu,
    tau.velocity.est < 1.5,
    region != 'atlantica'),
  method = 'REML')

pred.tv.f <- tibble(forest = seq_min_max(tapirs.lu$forest, n = 100),
  water = mean(tapirs.lu$water, na.rm = TRUE))
pred.tv.f <-
  bind_cols(pred.tv.f, predict(m.tauv.2, newdata = pred.tv.f, se.fit = TRUE,
    terms = 's(forest)')) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

pred.tv.w <- tibble(forest = mean(tapirs.lu$forest, na.rm = TRUE),
  water = seq_min_max(tapirs.lu$water, n = 100))
pred.tv.w <-
  bind_cols(pred.tv.w, predict(m.tauv.2, newdata = pred.tv.w, se.fit = TRUE,
    terms = 's(water)')) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))
```

```

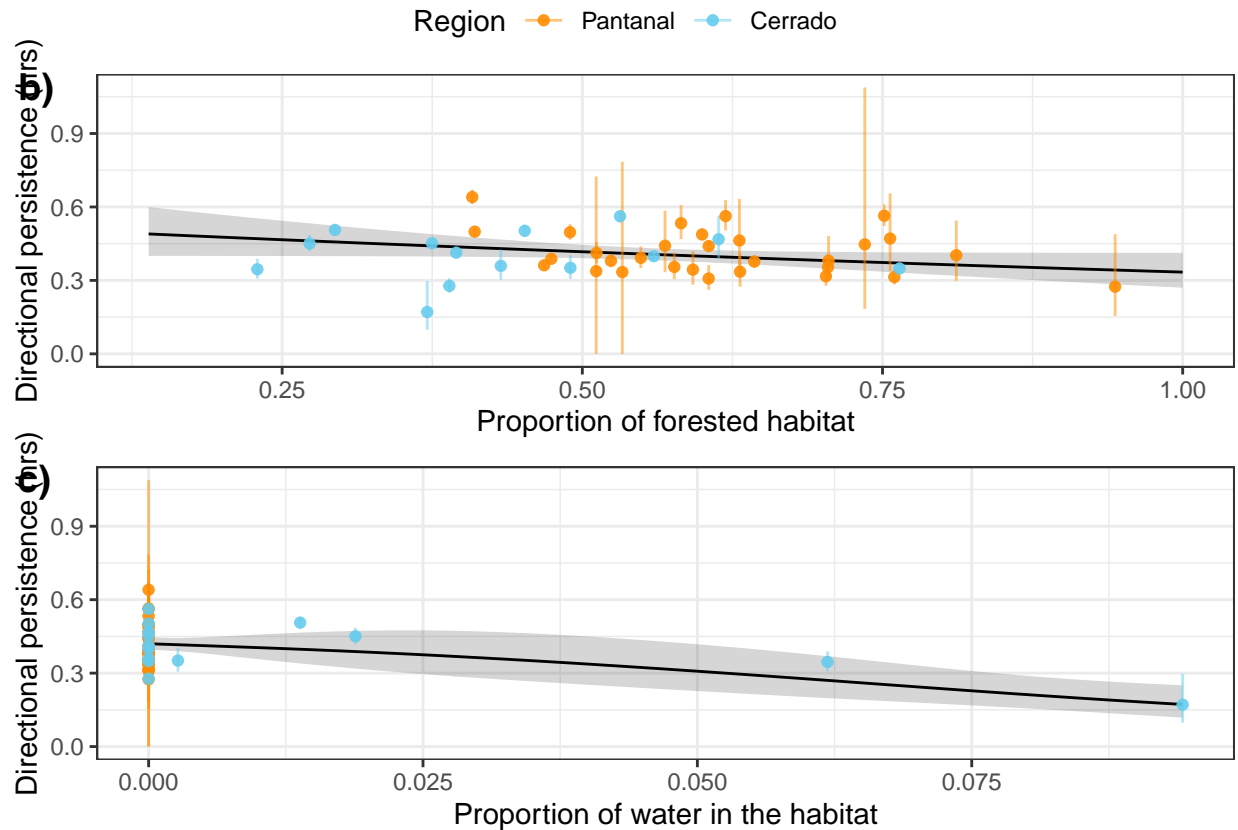
p.forest <-
  ggplot() +
    geom_ribbon(aes(forest, ymin = lwr, ymax = upr), pred.tv.f, alpha = 0.2) +
    geom_line(aes(forest, est), pred.tv.f) +
    geom_segment(aes(x = forest, xend = forest, y = tau.velocity.low,
                     yend = tau.velocity.high, color = region.lab),
                 filter(tapirs.lu,
                        tau.velocity.est < 1.5,
                        region != 'atlantica'), lwd = 0.5, alpha = 0.5) +
    geom_point(aes(forest, tau.velocity.est, color = region.lab),
               filter(tapirs.lu, tau.velocity.est < 1.5,
                      region != 'atlantica'), alpha = 0.9) +
    scale_color_manual('Region', values = pal[2:3]) +
    labs(x = 'Proportion of forested habitat',
         y = 'Directional persistence (hrs)') +
    theme(legend.position = 'none')

p.water <-
  ggplot() +
    geom_ribbon(aes(water, ymin = lwr, ymax = upr), pred.tv.w, alpha = 0.2) +
    geom_line(aes(water, est), pred.tv.w) +
    geom_segment(aes(x = water, xend = water, y = tau.velocity.low,
                     yend = tau.velocity.high, color = region.lab),
                 filter(tapirs.lu,
                        tau.velocity.est < 1.5,
                        region != 'atlantica'), lwd = 0.5, alpha = 0.5) +
    geom_point(aes(water, tau.velocity.est, color = region.lab),
               filter(tapirs.lu, tau.velocity.est < 1.5,
                      region != 'atlantica'), alpha = 0.9) +
    scale_color_manual('Region', values = pal[2:3]) +
    labs(x = 'Proportion of water in the habitat',
         y = 'Directional persistence (hrs)') +
    theme(legend.position = 'none')

plot_grid(get_legend(p.forest + theme(legend.position = 'top')),
          p.forest, p.water,
          ncol = 1, rel_heights = c(0.2, 1, 1, 1),
          labels = c(NA, 'b', 'c'))

```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```



Finally, as requested by reviewer #2, we can perform the regression shown in panel 5c without the zeros. However, since this reduces the data set to only 5 data points, we are forced to reduce the model's terms to linear or remove `s(forest)`. Since removing the zeros drastically changes the dataset, we decided to fit the dataset to the simple model with `s(water)` only.

```
tapirs.lu.w <- filter(tapirs.lu,
  tau.velocity.est < 1.5, # remove the outlier
  water > 0) # remove all instances of water == 0
nrow(tapirs.lu.w)

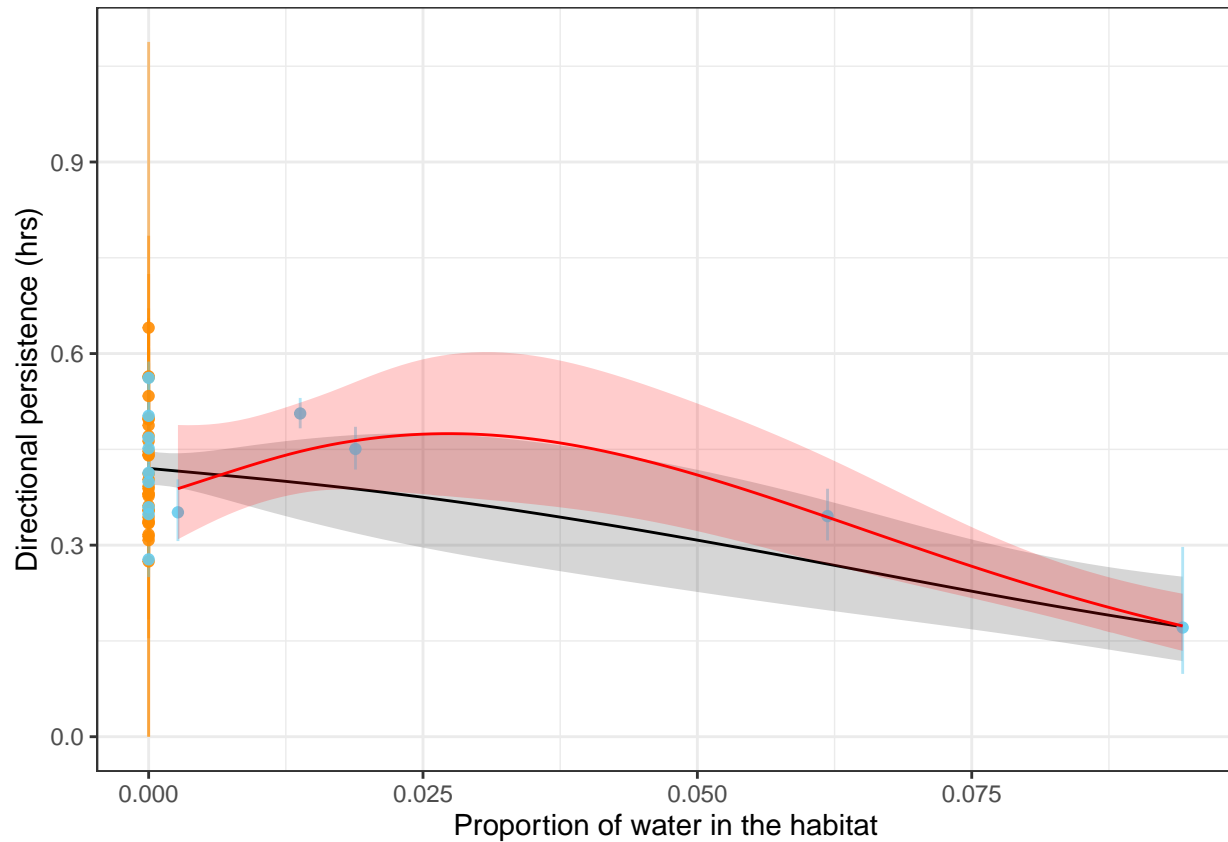
## [1] 5

m.tauv.w <- gam(tau.velocity.est ~ s(water, k = 5),
  family = Gamma('log'),
  data = tapirs.lu.w,
  method = 'REML')

pred.tv.w <- tibble(water = seq_min_max(tapirs.lu.w$water, n = 100))
pred.tv.w <- bind_cols(pred.tv.w,
  predict(m.tauv.w, newdata = pred.tv.w, se.fit = TRUE,
    terms = 's(water)')) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

p.water +
  geom_ribbon(aes(water, ymin = lwr, ymax = upr), pred.tv.w, alpha = 0.2,
```

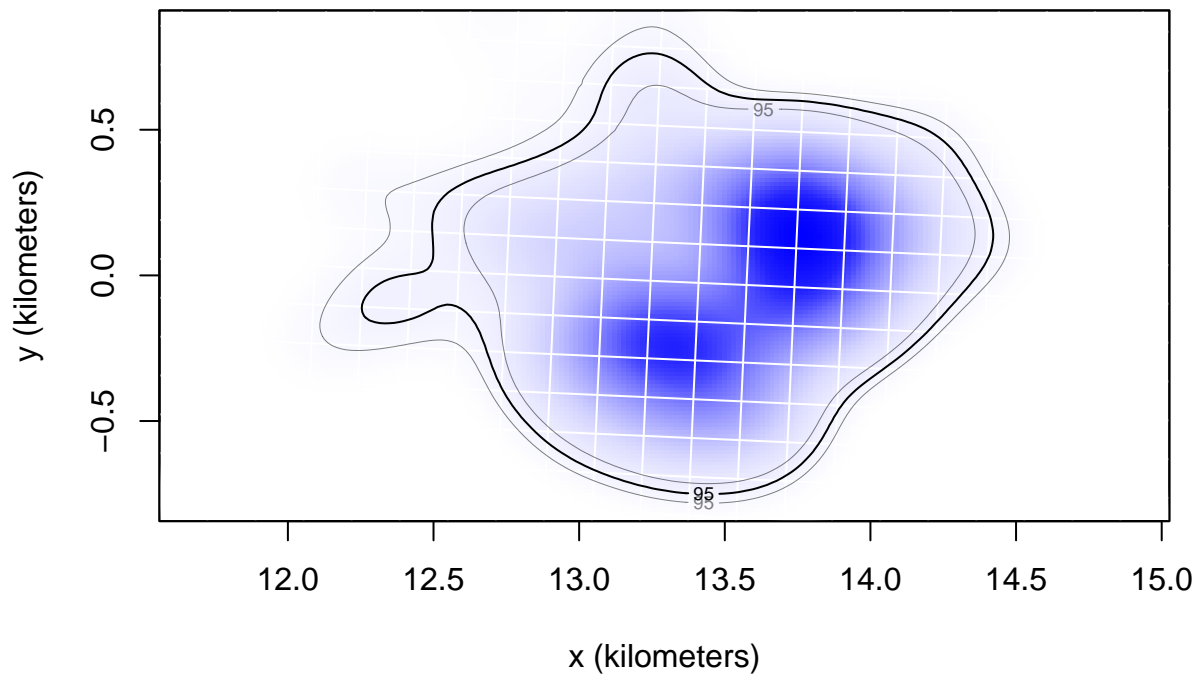
```
fill = 'red') +  
geom_line(aes(water, est), pred.tv.w, color = 'red')
```



Since the 95% confidence intervals of the original regression line (black) and the one fit to non-zero data (red) overlap, we have insufficient evidence to conclude that the presence of the zeros affected the model significantly.

Variation in movement across biomes and gradients of human disturbance (figure 6)

```
# import ml-HFI raster  
hfi.raster <- raster('../data/hfi-layers/ml_hfi_v1_2019.nc')  
  
## Loading required namespace: ncdf4  
  
# extract mean hfi and hr areas  
plot(tapirs$akde[[5]])
```



```
extract(hfi.raster, as.sf(tapirs$akde[[5]])) # [[1]]=lwr, [[2]]=est, [[3]]=upr

## [[1]]
##
## 0.2280571
##
## [[2]]
## [1] 0.1456574 0.2280571
##
## [[3]]
## [1] 0.1456574 0.2280571 0.1121567

tapirs <-
  tapirs %>%
  mutate(region = factor(region), # need factors for GAMs
         region.lab = if_else(region.lab == 'Mata Atlantica', # translate
                              true = 'Atlantic forest',
                              false = region.lab) %>%
         factor(levels = c('Atlantic forest', 'Pantanal', 'Cerrado')),
         hfi.mean = map_dbl(1:N,
                           function(i)
                             extract(hfi.raster,
                                      as.sf(akde[[i]]))[[2]] %>% # take estimate
                             mean(na.rm = TRUE)),
         hr.size = map_dbl(akde, function(a) summary(a)$CI[2]))
# warning on change of projection is ok (suppressed)
```

```

# hfi on hr size
m.hr.0 <- gam(hr.size ~ hfi.mean, # hfi.mean is linear on the link (log) scale
              family = Gamma('log'),
              data = tapirs,
              method = 'REML')
m.hr.1 <- gam(hr.size ~ s(hfi.mean), # allow hfi.mean to be smooth
              family = Gamma('log'),
              data = tapirs,
              method = 'REML')
m.hr.2 <- gam(hr.size ~ region + s(hfi.mean), # different intercept per region
              family = Gamma('log'),
              data = tapirs,
              method = 'REML')
m.hr.3 <- gam(hr.size ~ s(hfi.mean, by = region), # different smooth per region
              family = Gamma('log'),
              data = tapirs,
              method = 'REML')
m.hr.4 <- gam(hr.size ~ region + s(hfi.mean, by = region),
              family = Gamma('log'),
              data = tapirs,
              method = 'REML')

# accounting for effect of region does not improve the model fit
AIC(m.hr.0, m.hr.1, m.hr.2, m.hr.3, m.hr.4)

```

```

##           df      AIC
## m.hr.0 3.000000 444.6443
## m.hr.1 4.162897 444.5367
## m.hr.2 6.056456 447.4223
## m.hr.3 5.000777 444.0914
## m.hr.4 8.604185 447.7827

```

```
summary(m.hr.0)
```

```

##
## Family: Gamma
## Link function: log
##
## Formula:
## hr.size ~ hfi.mean
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0523     0.6760   3.036 0.00334 **
## hfi.mean      -0.3807     3.1431  -0.121 0.90393
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = -0.0138   Deviance explained = 0.0459%
## -REML = 219.04   Scale est. = 2.8666    n = 74

```

```

# regression plot
pred.hr <- tibble(hfi.mean = seq(0.003, 0.31, length.out = 400))
pred.hr <- bind_cols(pred.hr,

```

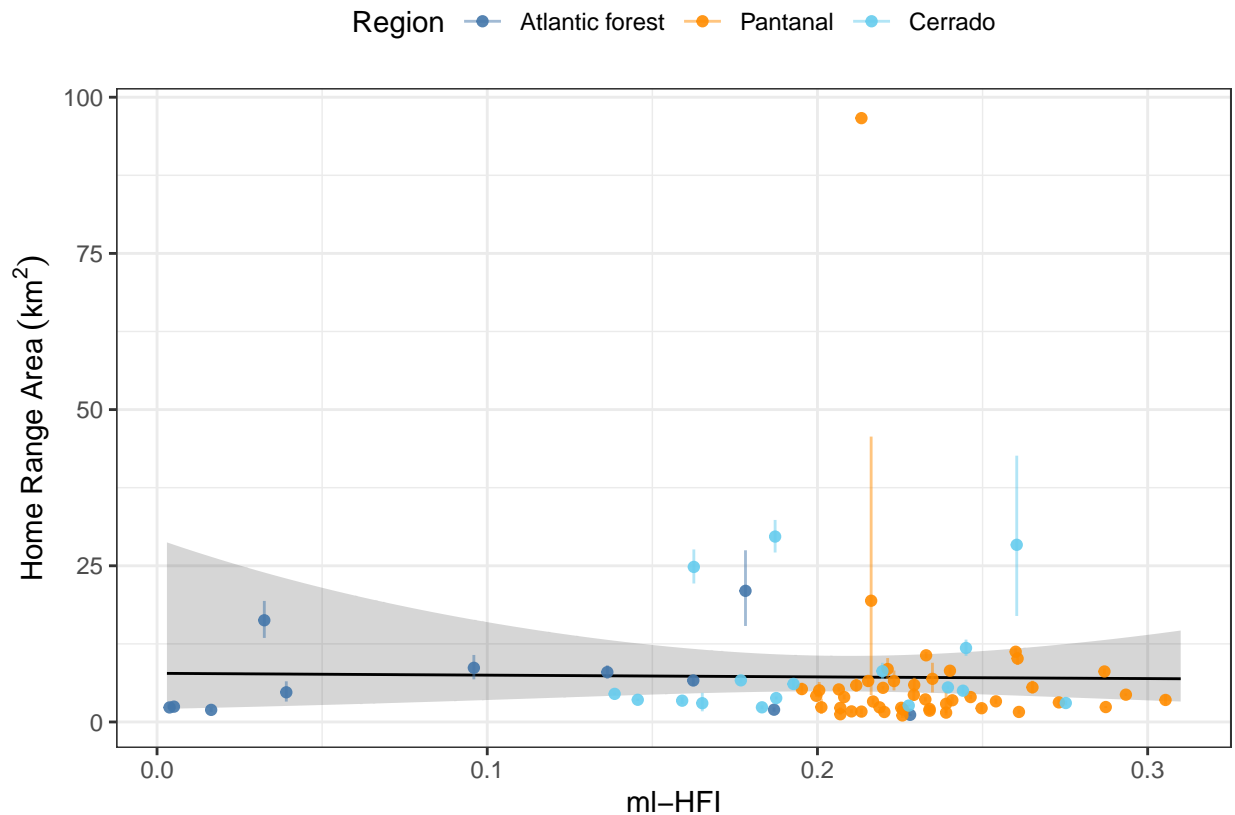


```

      predict(m.hr.0, newdata = pred.hr, se.fit = TRUE)) %>%
mutate(est = exp(fit),
      lwr = exp(fit - 1.96 * se.fit),
      upr = exp(fit + 1.96 * se.fit))

ggplot() +
  geom_ribbon(aes(hfi.mean, ymin = lwr, ymax = upr), pred.hr, alpha = 0.2) +
  geom_line(aes(hfi.mean, est), pred.hr) +
  geom_segment(aes(x = hfi.mean, xend = hfi.mean, y = area.low,
                  yend = area.high, color = region.lab), tapirs, lwd = 0.5,
              alpha = 0.5) +
  geom_point(aes(hfi.mean, hr.size, color = region.lab), tapirs, alpha = 0.9) +
  scale_color_manual('Region', values = pal[1:3]) +
  labs(x = 'ml-HFI', y = expression('Home Range Area'~(km^2))) +
  theme(legend.position = 'top')

```



```

# hfi on average speed
m.sp.0 <- gam(speed.est ~ s(hfi.mean), # shrunk a linear term by REML (edf == 1)
              family = Gamma('log'),
              data = tapirs,
              method = 'REML')
m.sp.1 <- gam(speed.est ~ region + s(hfi.mean),
              family = Gamma('log'),
              data = tapirs,
              method = 'REML')
m.sp.2 <- gam(speed.est ~ s(hfi.mean, by = region),

```

```

    family = Gamma('log'),
    data = tapirs,
    method = 'REML')
m.sp.3 <- gam(speed.est ~ region + s(hfi.mean, by = region),
    family = Gamma('log'),
    data = tapirs,
    method = 'REML')

# accounting for effect of region does not improve the model fit
AIC(m.sp.0, m.sp.1, m.sp.2, m.sp.3)

##           df      AIC
## m.sp.0 3.000217 312.5687
## m.sp.1 5.000033 315.1423
## m.sp.2 5.001660 316.2714
## m.sp.3 6.000541 317.2902

summary(m.sp.0)

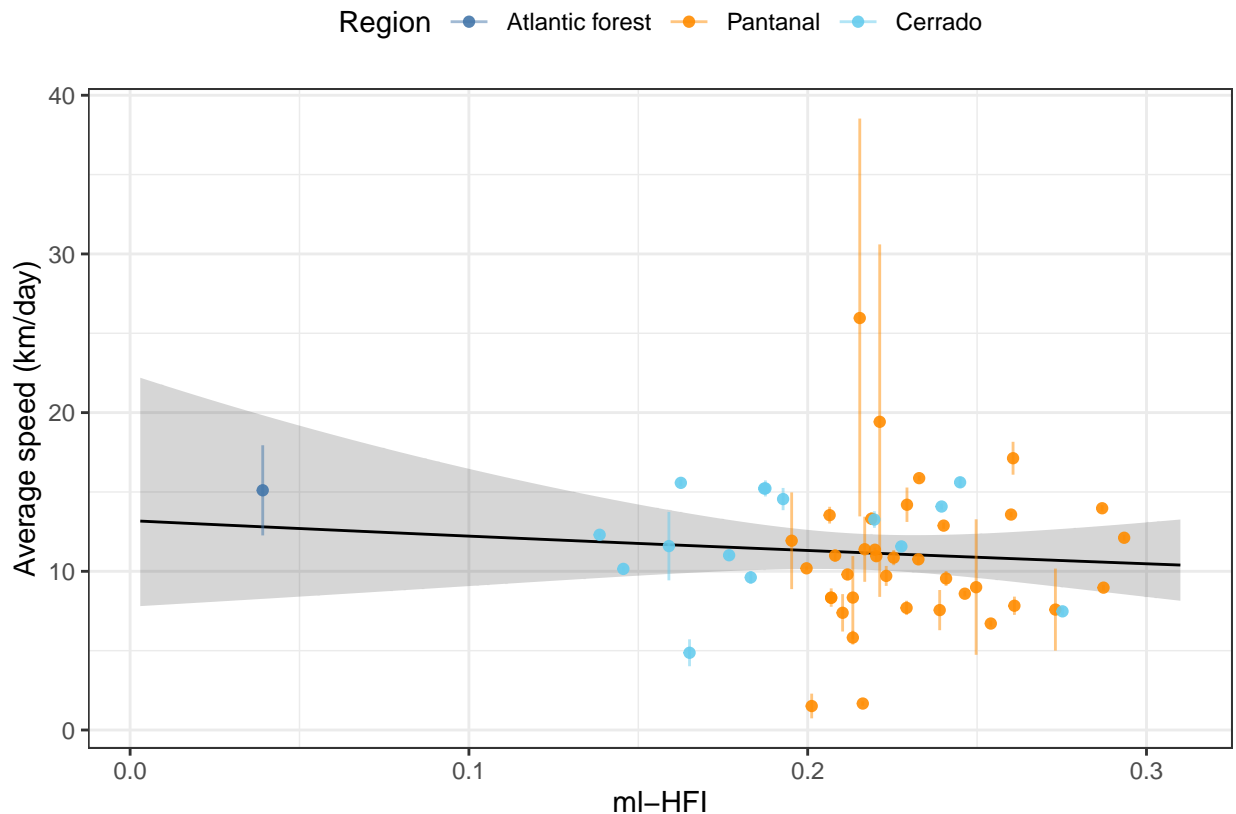
##
## Family: Gamma
## Link function: log
##
## Formula:
## speed.est ~ s(hfi.mean)
##
## Parametric coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.41271    0.05133     47  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df    F p-value
## s(hfi.mean)  1      1 0.397  0.532
##
## R-sq.(adj) = -0.0099  Deviance explained = 0.631%
## -REML = 157.17  Scale est. = 0.13965    n = 53

# regression plot
pred.sp <-
  bind_cols(dplyr::select(pred.hr, hfi.mean),
    predict(m.sp.0, newdata = pred.hr, se.fit = TRUE)) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

p3 <-
  ggplot(tapirs) +
  geom_ribbon(aes(hfi.mean, ymin = lwr, ymax = upr), pred.sp, alpha = 0.2) +
  geom_line(aes(hfi.mean, est), pred.sp) +
  geom_segment(aes(x = hfi.mean, xend = hfi.mean, y = speed.low,
    yend = speed.high, color = region.lab), lwd = 0.5,
    alpha = 0.5) +
  geom_point(aes(hfi.mean, speed.est, color = region.lab), alpha = 0.9)+

```

```
scale_color_manual('Region', values = pal[1:3]) +
labs(x = 'ml-HFI', y = 'Average speed (km/day)') +
theme(legend.position = 'top'); p3
```



As requested by reviewer #2, we can re-fit the regression model without the Atlantic forest data to show the data point (0.039, 15.1) has little effect on the conclusion drawn from the model (i.e. no significant effect of ml-HFI on average speed). The model without the Atlantic forest data point is indicated in red:

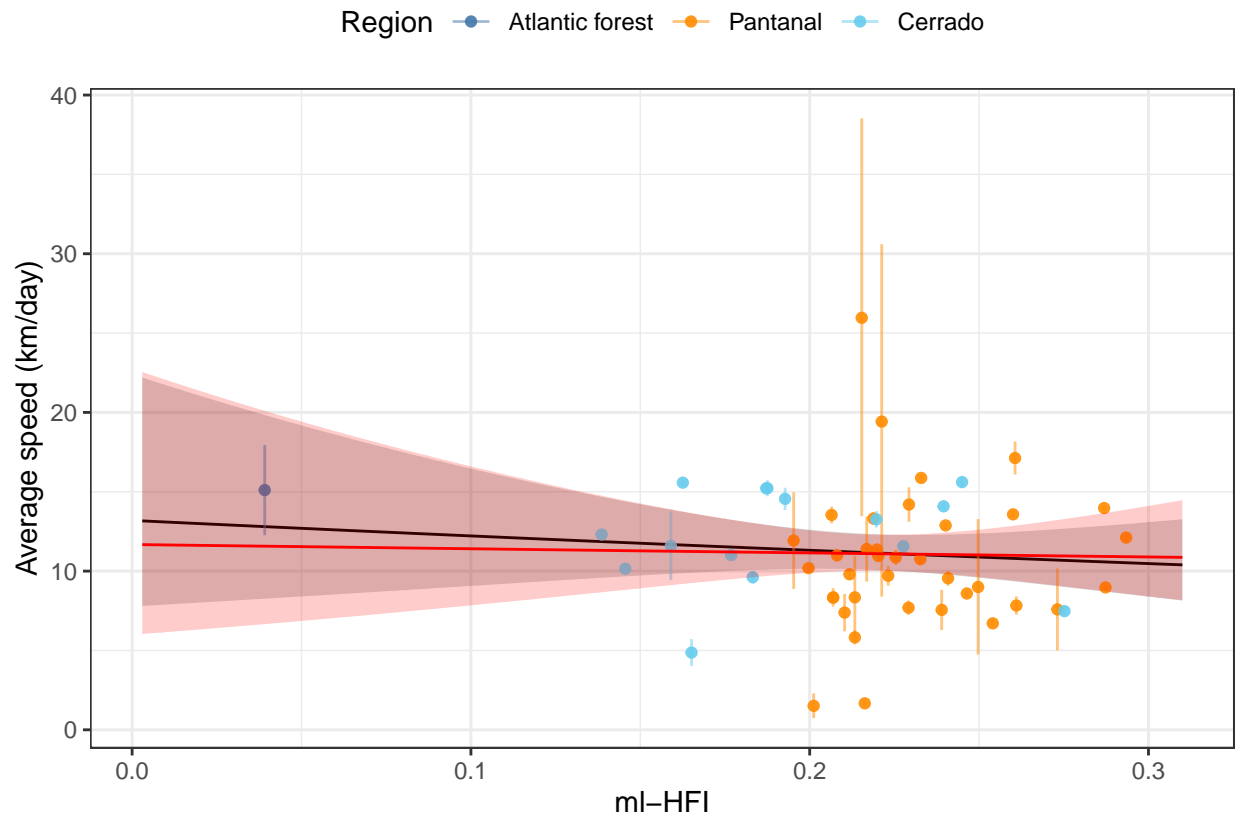
```
# without Mata Atlantica data
m.sp.0.a <- gam(speed.est ~ hfi.mean,
  family = Gamma('log'),
  data = filter(tapirs, region != 'atlantica'),
  method = 'REML')

pred.sp.a <-
  bind_cols(dplyr::select(pred.hr, hfi.mean),
    predict(m.sp.0.a, newdata = pred.hr, se.fit = TRUE)) %>%
  mutate(est = exp(fit),
    lwr = exp(fit - 1.96 * se.fit),
    upr = exp(fit + 1.96 * se.fit))

p3 +
  geom_ribbon(aes(hfi.mean, ymin = lwr, ymax = upr), pred.sp.a, alpha = 0.2,
    fill = 'red') +
  geom_line(aes(hfi.mean, est), pred.sp.a, color = 'red')
```

```
## Warning: Removed 21 rows containing missing values (geom_segment).
```

Warning: Removed 21 rows containing missing values (geom_point).



- new figures 5 and 6
- dredge()

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.1252 LC_CTYPE=English_Canada.1252
## [3] LC_MONETARY=English_Canada.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] gratia_0.6.9300 cowplot_1.1.1  ggplot2_3.3.5  raster_3.5-2
## [5] sp_1.4-6         sf_1.0-4       MuMIn_1.43.17  tidyr_1.1.4
## [9] ctmr_0.6.1       mgcv_1.8-38    nlme_3.1-153   purrr_0.3.4
## [13] dplyr_1.0.7
##
## loaded via a namespace (and not attached):
## [1] splines_4.1.2      Bessel_0.6-0      assertthat_0.2.1
## [4] expm_0.999-6       statmod_1.4.36    highr_0.9
## [7] stats4_4.1.2       yaml_2.2.1        numDeriv_2016.8-1.1
## [10] pillar_1.6.4       lattice_0.20-45   glue_1.5.1
## [13] digest_0.6.29      colorspace_2.0-2  htmltools_0.5.2
## [16] Matrix_1.3-4       pkgconfig_2.0.3   patchwork_1.1.1
## [19] scales_1.1.1       terra_1.4-20      tibble_3.1.6
## [22] proxy_0.4-26       gmp_0.6-2.1       generics_0.1.1
## [25] farver_2.1.0       ellipsis_0.3.2    withr_2.4.3
## [28] Rmpfr_0.8-7        cli_3.1.0         magrittr_2.0.1
## [31] crayon_1.4.2       mvnfast_0.2.7     evaluate_0.14
## [34] ncd4_1.18          fansi_0.5.0       class_7.3-19
## [37] tools_4.1.2        lifecycle_1.0.1   stringr_1.4.0
## [40] munsell_0.5.0      compiler_4.1.2    e1071_1.7-9
## [43] rlang_0.4.12       classInt_0.4-3    units_0.7-2
## [46] grid_4.1.2         rstudioapi_0.13   labeling_0.4.2
## [49] rmarkdown_2.11     gtable_0.3.0      codetools_0.2-18
## [52] DBI_1.1.1          R6_2.5.1          knitr_1.36
## [55] rgdal_1.5-27       fastmap_1.1.0     rgeos_0.5-8
## [58] utf8_1.2.2         KernSmooth_2.23-20 stringi_1.7.6
## [61] parallel_4.1.2     Rcpp_1.0.7        vctrs_0.3.8
## [64] tidyselect_1.1.1   xfun_0.28
```