# # Code for finding breakpoint for when an animal’s home-range size plateaus, regardless of adding additional tracking data

**# Based on the code on this website:**

# <http://stats.stackexchange.com/questions/19772/estimating-the-break-point-in-a-broken-stick-piecewise-linear-model-with-rando/19777#19777>

# Before you run this code, delete any unnecessary columns from your spreadsheet

# remove previous objects present in the workspace in R

rm(list=ls())

library(lme4)

#load your data & rename it ‘dat’

dat <- read.csv("~/your file name & location.csv")

## ### cMCP100% ###

# Here we calculate the breakpoint (in terms of number of months) for cumulative 100% MCP (minimum convex polygon).

### # 1st try

# we guestimate a breakpoint of 15 months, i.e. that home-range size will plateau after 15 months of tracking.

# in this example we searched for a breakpoint that fell within the range zero to 30 months

bp = 15

b1 <- function(x, bp) ifelse(x < bp, bp - x, 0)

b2 <- function(x, bp) ifelse(x < bp, 0, x - bp)

foo <- function(bp)

{

mod <- lmer(cMCP100 ~ Subspecies + AgeClass + Sex + b1(MonthOfStudy, bp) + b2(MonthOfStudy, bp) + (b1(MonthOfStudy, bp) + b2(MonthOfStudy, bp)|Bird) + (1|DutyCycle) + (1|No\_Fixes\_Monthly), data = dat)

deviance(mod)

}

search.range <- c(0,30)

foo.opt <- optimize(foo, interval = search.range)

bp <- foo.opt$minimum

bp

# [1] 8.081593

foo.root <- function(bp, tgt) + {foo(bp) - tgt}

tgt <- foo.opt$objective + qchisq(0.95,1)

lb95 <- uniroot(foo.root, lower=search.range[1], upper=bp, tgt=tgt)

ub95 <- uniroot(foo.root, lower=bp, upper=search.range[2], tgt=tgt)

ub95$root

# [1] 8.216483

lb95$root

# [1] 8.081593

# Our results for bp, ub95$root, and lb95$root tell us that the months up to and including 8 are lower slope,

# and months 9 and above are upper slope.

### # 2nd try

# this time we guestimate a breakpoint of 20 months, i.e. that home-range size will plateau after 20 months of tracking.

# in this example we searched for a breakpoint that fell within the range zero to 48 months (we had up to 48 months of tracking data for each bird)

library(lme4)

bp = 20

b1 <- function(x, bp) ifelse(x < bp, bp - x, 0)

b2 <- function(x, bp) ifelse(x < bp, 0, x - bp)

foo <- function(bp)

{

mod <- lmer(cMCP100 ~ Subspecies + AgeClass + Sex + b1(MonthOfStudy, bp) + b2(MonthOfStudy, bp) + (b1(MonthOfStudy, bp) + b2(MonthOfStudy, bp)|Bird) + (1|DutyCycle) + (1|No\_Fixes\_Monthly), data = dat)

deviance(mod)

}

search.range <- c(0,48)

foo.opt <- optimize(foo, interval = search.range)

bp <- foo.opt$minimum

bp

# [1] 8.045798

foo.root <- function(bp, tgt) + {foo(bp) - tgt}

tgt <- foo.opt$objective + qchisq(0.95,1)

lb95 <- uniroot(foo.root, lower=search.range[1], upper=bp, tgt=tgt)

ub95 <- uniroot(foo.root, lower=bp, upper=search.range[2], tgt=tgt)

ub95$root

# [1] 8.06155

lb95$root

# [1] 8.045798

# Again, our results for bp, ub95$root, and lb95$root tell us that the months up to and including 8 are lower slope,

# and that months 9 and above are upper slope.

## # This can be repeated for other measures of home-range size estimates, by replacing the response variable in your model

# e.g. replace cMCP100 with cKDE50 or cKDE95, etc.

# You can test whether the gradients calculated for lower and upper slopes fit well by finding R-squared and p.

### Getting slopes for below and above the breaking point…

# Split your data into months 1 to 8 (this will be ‘dat’ for lower slope)

# and months 9 and up (this will be ‘dat’ for upper slope).

# Then load your data each time & rename it ‘dat’ for the sake of brevity.

# note the R-squared values in the summaries below, and the p-values for the lines fit to the lower and upper slopes:

mod1a <- lmer(cMCP100\_LowerSlope ~ Subspecies + AgeClass + Sex + (1|Bird) + (1|DutyCycle) + (1|No\_Fixes\_Monthly), data = dat)

mod1a

summary(mod1a)

mod1b <- lmer(cMCP100\_UpperSlope ~ Subspecies + AgeClass + Sex + (1|Bird) + (1|DutyCycle) + (1|No\_Fixes\_Monthly), data = dat)

mod1b

summary(mod1b)