Mountain Goat RSF Metadata

Author: McCrea Cobb

Last updated: 9/15/2017

/KodiakGoatRSF/Analysis

ImportFormat.R
Imports GPS collar data from ATS and Telonics collars. Reformats the data, removes erronious locations, and merges it into a single dataframe. Creates a spatial dataframe and plots it as a visual check for errors.

FormatSpatailCov.R
Imports and formats spatial covariate shapefiles and rasters. Scales covariates when appropriate. Creates binomial rasters for each habitat class in the land cover classification (LCC). Creates and saves a raster stack of the formatted habitat covariate rasters. Imports and plots the raster stack. Saves plot as a pdf.

ExtractCovVelox.R
Defines "available" fixes for standard RSF modeling by creating a 99% kernel density spatial polygon, using all fixes. Generates a random sample of fixes within the polygon and combines these data to the "used" GPS location dataframe, called dfRSF. Imports a raster stack of habitat covariates. Extract the covariate data from the raster stack to the used/available data. Export the result as a shapefile.

RSFModel.R
Examines resource selection based on mixed effects logistic regression (CollarID as a random effect). Checks for covariance among the habitat covariates. Saves the resulting table as a .csv. Splits the used/available data frame into testing and training dataframes. Creates a list of terrian models, based on topography. Checks models for overdispersion. Runs AIC model selection to determine the best model. Uses this as the base model for a second set of candidate models based on habitat covariates. Runs AIC model selection to determine the best model. Checks models for overdispersion. Creates a table of parameter estimates and CIs from the top model. Adds predicted values (based on the top model) to the used/available data frame. Evaluates the goodness of fit of the best model with Hosmer-Lemeshow Goodness of Fit (GOF) test.

ConditionalRSFmodel.R Examines resource selection based on mixed effects conditional logistic regression (CollarID as a random effect). Splits the used/available data frame into testing and training dataframes. Creates a dataset of available fixes based on the average distance moved during consequetive fixes. To do this, first creates a trajectory object using the "used" GPS locations that includes the length of each move, the time interval between successive relocations and other values. Creates a table that summarized these values by CollarID.