Example script for VAST for spatio-temporal analysis of single-species catch-rate data

James Thorson

October 10, 2016

Contents

```
## package 'pander' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\James.Thorson\AppData\Local\Temp\RtmpE7YgTm\downloaded_packages
```

1 Overview

This tutorial will walk through a simple example of how to use `VAST` for estimating single-species abundance indices, distribution shifts, and range expansion.

2 Getting started

To install TMB on a windows machine, we need to first install Rtools. During the installation, please select the option to have Rtools included in your system path. On other operating systems, it is not necessary to install Rtools. We then install `VAST`

```
devtools::install_github("james-thorson/VAST")
devtools::install_github("james-thorson/utilities")
```

Next load libraries.

```
library(TMB)               # Can instead load library(TMBdebug)
library(VAST)
```

2.1 Further information

If you have further questions after reading this tutorial, please explore the GitHub repo mainpage, wiki, and glossary. Also please explore the R help files, e.g., `?Data_Fn` for explanation of data inputs, or `?Param_Fn` for explanation of parameters.

2.2 Related tools

Related tools for spatio-temporal fisheries analysis are currently housed at www.FishStats.org. These include SpatialDeltaGLMM, a single-species antecedent of VAST, and www.FishViz.org, a tool for visualizing single-species results using worldwide. `VAST` and `SpatialDeltaGLMM` both use continuous integration to confirm that they give identical estimates when applied to single-species data.

2.3 How to cite VAST

`VAST` has involved many publications for developing individual features. If using `VAST`, please read and cite:

```
citation("VAST")
```

```
##
## Please cite 2016 (ICES J. Mar. Sci. J.
## Cons.) if using the package; 2016 (Glob.
## Ecol. Biogeogr) if exploring factor
## decomposition of spatio-temporal variation;
## 2015 (ICES J. Mar. Sci. J. Cons.) if
## calculating an index of abundance; 2016
## (Methods Ecol. Evol.) if using the
## center-of-gravity metric; 2016 (Fish. Res.)
## if using the bias-correction feature; 2016
## (Proc R Soc B) if using the
## effective-area-occupied metric.
##
##   Thorson, J.T., and Barnett, L.A.K. In
##   press. Comparing estimates of abundance
##   trends and distribution shifts using
##   single- and multispecies models of fishes
##   and biogenic habitat. ICES J. Mar. Sci. J.
##   Cons
##
##   Thorson, J.T., Ianelli, J.N., Larsen, E.,
##   Ries, L., Scheuerell, M.D., Szuwalski, C.,
##   and Zipkin, E. 2016. Joint dynamic species
##   distribution models: a tool for community
##   ordination and spatiotemporal monitoring.
##   Glob. Ecol. Biogeogr. 25(9): 1144-1158.
##   doi:10.1111/geb.12464. url:
##   http://onlinelibrary.wiley.com/doi/10.1111/geb.12464/abstract
##
##   Thorson, J.T., Shelton, A.O., Ward, E.J.,
##   Skaug, H.J., 2015. Geostatistical
##   delta-generalized linear mixed models
##   improve precision for estimated abundance
##   indices for West Coast groundfishes. ICES
##   J. Mar. Sci. J. Cons. 72(5), 1297-1310.
##   doi:10.1093/icesjms/fsu243. URL:
##   http://icesjms.oxfordjournals.org/content/72/5/1297
##
##   Thorson, J.T., and Kristensen, K. 2016.
##   Implementing a generic method for bias
##   correction in statistical models using
##   random effects, with spatial and
##   population dynamics examples. Fish. Res.
##   175: 66-74.
##   doi:10.1016/j.fishres.2015.11.016. url:
##   http://www.sciencedirect.com/science/article/pii/S0165783615301399
##
##   Thorson, J.T., Pinsky, M.L., Ward, E.J.,
```

```
##   2016. Model-based inference for estimating
##   shifts in species distribution, area
##   occupied, and center of gravity. Methods
##   Ecol. Evol. 7(8), 990-1008.
##   doi:10.1111/2041-210X.12567.  URL:
##   http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12567/full
##
##   Thorson, J.T., Rindorf, A., Gao, J.,
##   Hanselman, D.H., and Winker, H. 2016.
##   Density-dependent changes in effective
##   area occupied for sea-bottom-associated
##   marine fishes. Proc R Soc B 283(1840):
##   20161853. doi:10.1098/rspb.2016.1853. URL:
##   http://rspb.royalsocietypublishing.org/content/283/1840/20161853.
```

and also browse the GitHub list of papers

3 Settings

First chose an example data set for this script, as archived with package

```
Data_Set = c("Chatham_rise_hake", "Iceland_cod", "WCGBTS_canary",
    "GSL_american_plaice", "BC_pacific_cod", "EBS_pollock",
    "GOA_Pcod", "GOA_pollock", "GB_spring_haddock",
    "GB_fall_haddock", "SAWC_jacopever", "Aleutian_islands_POP")[12]
```

Next use latest version for CPP code

```
Version = "VAST_v2_0_0"
```

3.1 Spatial settings

The following settings define the spatial resolution for the model, and whether to use a grid or mesh approximation

```
Method = c("Grid", "Mesh", "Spherical_mesh")[2]
grid_size_km = 25
n_x = c(100, 250, 500, 1000, 2000)[1] # Number of stations
Kmeans_Config = list( "randomseed"=1, "nstart"=100, "iter.max"=1e3 )
```

3.2 Model settings

The following settings define whether to include spatial and spatio-temporal variation, whether its autocorrelated, and whether there's overdispersion

```
FieldConfig = c(Omega1 = 1, Epsilon1 = 1, Omega2 = 1,
    Epsilon2 = 1)
RhoConfig = c(Beta1 = 0, Beta2 = 0, Epsilon1 = 0, Epsilon2 = 0)
OverdispersionConfig = c(Vessel = 0, VesselYear = 0)
ObsModel = c(2, 0)
```

3.3 Potential outputs

The following settings define what types of output we want to calculate

```
Options = c(SD_site_density = 0, SD_site_logdensity = 0,
    Calculate_Range = 1, Calculate_evenness = 0, Calculate_effective_area = 1,
    Calculate_Cov_SE = 0, Calculate_Synchrony = 0,
    Calculate_Coherence = 0)
```

3.4 Stratification for results

We also define any potential stratification of results, and settings specific to any case-study data set

```
# Default
if (Data_Set %in% c("GSL_american_plaice", "BC_pacific_cod",
    "EBS_pollock", "SAWC_jacopever", "Chatham_rise_hake",
    "Aleutian_islands_POP")) {
    strata.limits <- data.frame(STRATA = "All_areas")
}
# Specific (useful as examples)
if (Data_Set %in% c("WCGBTS_canary", "Sim")) {
    # In this case, it will calculate a coastwide
    # index, and also a separate index for each state
    # (although the state lines are approximate)
    strata.limits <- data.frame(STRATA = c("Coastwide",
        "CA", "OR", "WA"), north_border = c(49, 42,
        46, 49), south_border = c(32, 32, 42, 46),
        shallow_border = c(55, 55, 55, 55), deep_border = c(1280,
            1280, 1280, 1280))
    # Override default settings for vessels
    VesselConfig = c(Vessel = 0, VesselYear = 1)
}
if (Data_Set %in% c("GOA_Pcod", "GOA_pollock")) {
    # In this case, will calculating an unrestricted
    # index and a separate index restricted to west of
    # -140W
    strata.limits <- data.frame(STRATA = c("All_areas",
        "west_of_140W"), west_border = c(-Inf, -Inf),
        east_border = c(Inf, -140))
}
if (Data_Set %in% c("GB_spring_haddock", "GB_fall_haddock")) {
    # For NEFSC indices, strata must be specified as a
    # named list of area codes
    strata.limits = list(Georges_Bank = c(1130, 1140,
        1150, 1160, 1170, 1180, 1190, 1200, 1210, 1220,
        1230, 1240, 1250, 1290, 1300))
}
if (Data_Set %in% c("Iceland_cod")) {
    strata.limits = data.frame(STRATA = "All_areas")
    # Turn off all spatial, temporal, and
    # spatio-temporal variation in probability of
    # occurrence, because they occur almost everywhere
    FieldConfig = c(Omega1 = 0, Epsilon1 = 0, Omega2 = 1,
```

```
        Epsilon2 = 1)
    RhoConfig = c(Beta1 = 3, Beta2 = 0, Epsilon1 = 0,
        Epsilon2 = 0)
}
```

3.5 Derived objects

Depending on the case study, we define a `Region` used when extrapolating or plotting density estimates. If its a different data set, it will define `Region="Other"`, and this is a recognized level for all uses of `Region` (which attempts to define reasonable settings based on the location of sampling). For example `Data_Set="Iceland_cod"` has no associated meta-data for the region, so it uses `Region="Other"` by default.

```
Region = switch( Data_Set, "Chatham_rise_hake"="New_Zealand",
                 "WCGBTS_canary"="California_current",
                 "GSL_american_plaice"="Gulf_of_St_Lawrence",
                 "BC_pacific_cod"="British_Columbia",
                 "EBS_pollock"="Eastern_Bering_Sea",
                 "GOA_Pcod"="Gulf_of_Alaska",
                 "GOA_pollock"="Gulf_of_Alaska",
                 "GB_spring_haddock"="Northwest_Atlantic",
                 "GB_fall_haddock"="Northwest_Atlantic",
                 "SAWC_jacopever"="South_Africa",
                 "Aleutian_islands_POP"="Aleutian_Islands",
                 "Other")
```

3.6 Save settings

We then set the location for saving files.

```
DateFile = paste0(getwd(),'/VAST_output/')
  dir.create(DateFile)
```

I also like to save all settings for later reference, although this is not necessary.

```
Record = ThorsonUtilities::bundlelist(c("Data_Set",
    "Version", "Method", "grid_size_km", "n_x", "FieldConfig",
    "RhoConfig", "OverdispersionConfig", "ObsModel",
    "Kmeans_Config"))
save(Record, file = file.path(DateFile, "Record.RData"))
capture.output(Record, file = paste0(DateFile, "Record.txt"))
```

4 Prepare the data

4.1 Data-frame for catch-rate data

Depending upon the `Data_Set` chosen, we load archived data sets that are distributed with the package. Each archived data set is then reformatted to create a data-frame `Data_Geostat` with a standardized set of columns. For a new data set, the user is responsible for formatting `Data_Geostat` appropriately to match this format. We show the first six rows of `Data_Geostat` given that Data_Set = `Data_Set`.

Catch_KG	Year	Vessel	AreaSwept_km2	Lat	Lon	Pass
0	1991	missing	1	52.1	-176	0
252	1991	missing	1	52.5	-180	0
1395	1991	missing	1	52.5	-180	0
451	1991	missing	1	52	-180	0
102	1991	missing	1	52	-180	0
28.7	1991	missing	1	52.2	-180	0

4.2 Extrapolation grid

We also generate the extrapolation grid appropriate for a given region. For new regions, we use `Region="Other"`.

```
if (Region %in% c("California_current", "Eastern_Bering_Sea",
    "Gulf_of_Alaska", "Aleutian_Islands", "Northwest_Atlantic",
    "Gulf_of_St_Lawrence", "New_Zealand")) {
    Extrapolation_List = SpatialDeltaGLMM::Prepare_Extrapolation_Data_Fn(Region = Region,
        strata.limits = strata.limits)
}
if (Region == "British_Columbia") {
    Extrapolation_List = SpatialDeltaGLMM::Prepare_Extrapolation_Data_Fn(Region = Region,
        strata.limits = strata.limits, strata_to_use = c("HS",
            "QCS"))
}
if (Region == "South_Africa") {
    Extrapolation_List = SpatialDeltaGLMM::Prepare_Extrapolation_Data_Fn(Region = Region,
        strata.limits = strata.limits, region = "west_coast")
}
if (Region == "Other") {
    Extrapolation_List = SpatialDeltaGLMM::Prepare_Extrapolation_Data_Fn(Region = Region,
        strata.limits = strata.limits, observations_LL = Data_Geostat[,
            c("Lat", "Lon")], maximum_distance_from_sample = 15)
}
```

4.3 Derived objects for spatio-temporal estimation

And we finally generate the information used for conducting spatio-temporal parameter estimation, bundled in list `Spatial_List`

```
Spatial_List = SpatialDeltaGLMM::Spatial_Information_Fn(grid_size_km = grid_size_km,
    n_x = n_x, Method = Method, Lon = Data_Geostat[,
        "Lon"], Lat = Data_Geostat[, "Lat"], Extrapolation_List = Extrapolation_List,
    randomseed = Kmeans_Config[["randomseed"]], nstart = Kmeans_Config[["nstart"]],
    iter.max = Kmeans_Config[["iter.max"]], DirPath = DateFile,
    Save_Results = FALSE)
# Add knots to Data_Geostat
Data_Geostat = cbind(Data_Geostat, knot_i = Spatial_List$knot_i)
```

5 Build and run model

5.1 Build model

To estimate parameters, we first build a list of data-inputs used for parameter estimation. `Data_Fn` has some simple checks for buggy inputs, but also please read the help file `?Data_Fn`.

```
TmbData = Data_Fn(Version = Version, FieldConfig = FieldConfig,
    OverdispersionConfig = OverdispersionConfig, RhoConfig = RhoConfig,
    ObsModel = ObsModel, c_i = rep(0, nrow(Data_Geostat)),
    b_i = Data_Geostat[, "Catch_KG"], a_i = Data_Geostat[,
        "AreaSwept_km2"], v_i = as.numeric(Data_Geostat[,
        "Vessel"]) - 1, s_i = Data_Geostat[, "knot_i"] -
        1, t_i = Data_Geostat[, "Year"], a_xl = Spatial_List$a_xl,
    MeshList = Spatial_List$MeshList, GridList = Spatial_List$GridList,
    Method = Spatial_List$Method, Options = Options)
```

```
##    Omega1 Epsilon1   Omega2 Epsilon2
##         1        1        1        1
##     Vessel VesselYear
##         -1         -1
```

We then build the TMB object.

```
TmbList = Build_TMB_Fn(TmbData = TmbData, RunDir = DateFile,
    Version = Version, RhoConfig = RhoConfig, loc_x = Spatial_List$loc_x,
    Method = Method)
Obj = TmbList[["Obj"]]
```

5.2 Estimate fixed effects and predict random effects

Next, we use a gradient-based nonlinear minimizer to identify maximum likelihood estimates for fixed-effects

```
Opt = TMBhelper::Optimize(obj = Obj, lower = TmbList[["Lower"]],
    upper = TmbList[["Upper"]], getsd = TRUE, savedir = DateFile,
    bias.correct = FALSE)
```

Finally, we bundle and save output

```
Report = Obj$report()
Save = list("Opt"=Opt, "Report"=Report, "ParHat"=Obj$env$parList(Opt$par), "TmbData"=TmbData)
save(Save, file=paste0(DateFile,"Save.RData"))
```

6 Diagnostic plots

We first apply a set of standard model diagnostics to confirm that the model is reasonable and deserves further attention. If any of these do not look reasonable, the model output should not be interpreted or used.

6.1 Plot data

It is always good practice to conduct exploratory analysis of data. Here, I visualize the spatial distribution of data. Spatio-temporal models involve the assumption that the probability of sampling a given location is statistically independent of the probability distribution for the response at that location. So if sampling "follows" changes in density, then the model is probably not appropriate!

```
SpatialDeltaGLMM::Plot_data_and_knots(Extrapolation_List = Extrapolation_List,
    Spatial_List = Spatial_List, Data_Geostat = Data_Geostat,
    PlotDir = DateFile)
```

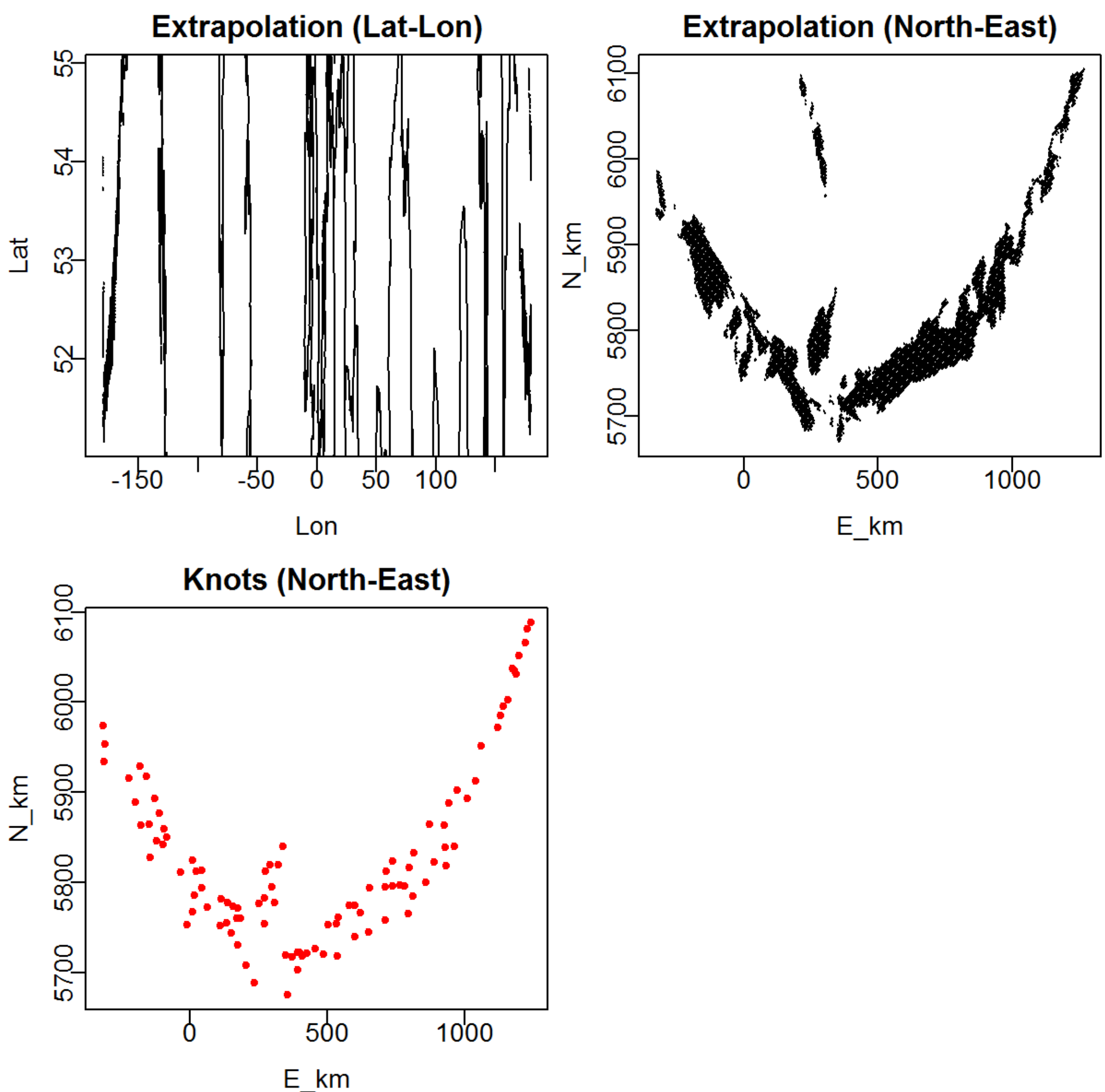

Figure 1: Spatial extent and location of knots

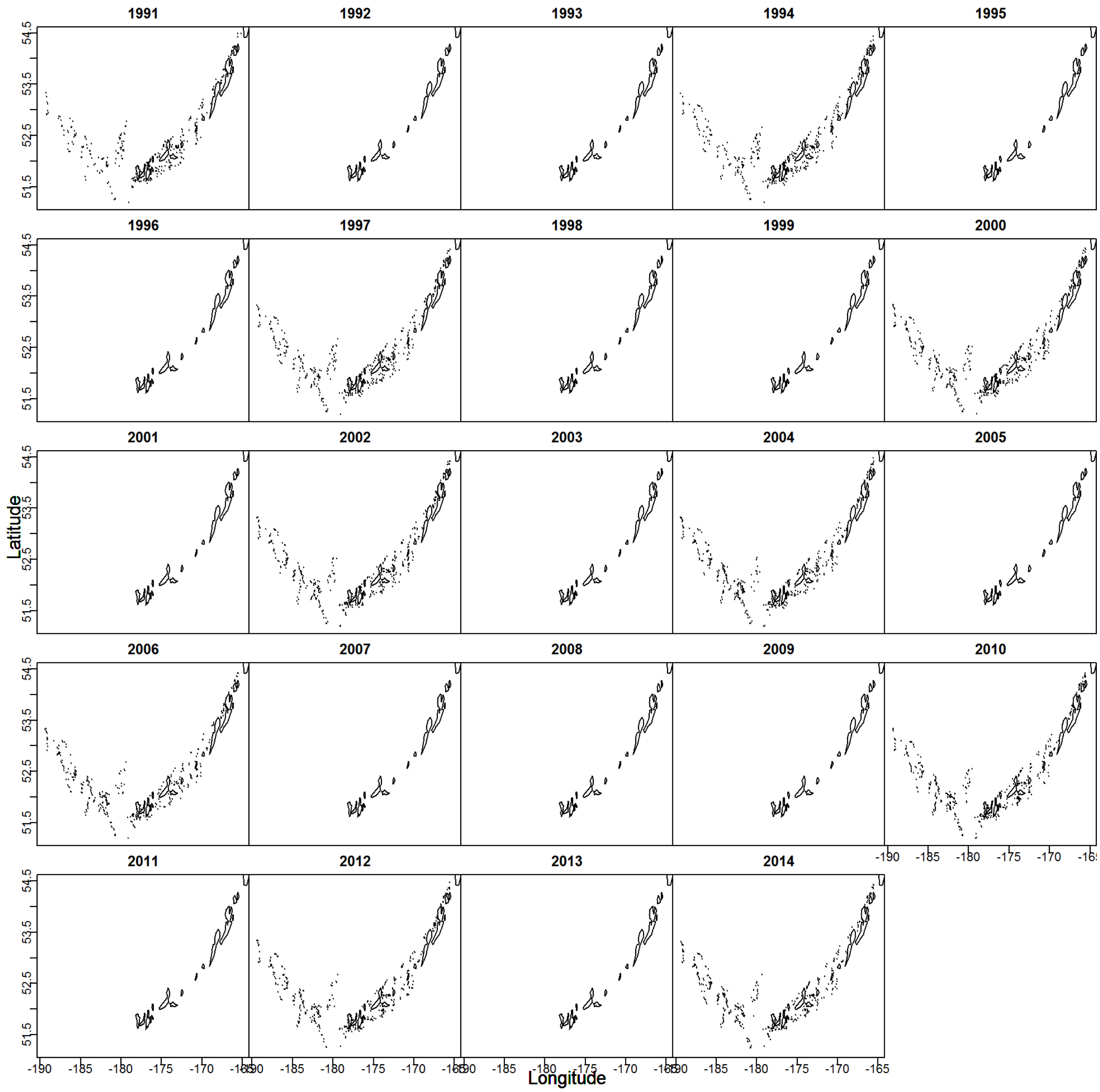

Figure 2: Spatial distribution of catch-rate data

6.2 Convergence

Here I print the diagnostics generated during parameter estimation, and I confirm that (1) no parameter is hitting an upper or lower bound and (2) the final gradient for each fixed-effect is close to zero. For explanation of parameters, please see `?Data_Fn`.

```
pander::pandoc.table( Opt$diagnostics[,c('Param','Lower','MLE','Upper','final_gradient')] )
```

Param	Lower	MLE	Upper	final_gradient
ln_H_input	-50	0.6577	50	6.648e-05
ln_H_input	-50	1.033	50	8.108e-05
beta1_ct	-50	0.8199	50	-1.744e-06
beta1_ct	-50	0.4746	50	7.807e-06
beta1_ct	-50	0.7243	50	-1.111e-05
beta1_ct	-50	0.8558	50	-2.513e-05
beta1_ct	-50	0.89	50	2.555e-05
beta1_ct	-50	0.9905	50	2.11e-05
beta1_ct	-50	1.025	50	-3.986e-06
beta1_ct	-50	0.945	50	-3.258e-06
beta1_ct	-50	1.55	50	3.961e-06
beta1_ct	-50	1.659	50	3.405e-06
L_omega1_z	-50	2.602	50	3.237e-05
L_epsilon1_z	-50	-0.3273	50	-9.758e-06
logkappa1	-6.313	-2.779	-0.4504	1.924e-05
beta2_ct	-50	8.605	50	9.396e-05
beta2_ct	-50	8.912	50	-3.108e-05
beta2_ct	-50	9.332	50	-5.679e-05
beta2_ct	-50	9.29	50	-1.624e-06
beta2_ct	-50	9.183	50	-3.295e-05
beta2_ct	-50	9.74	50	-0.000127
beta2_ct	-50	9.734	50	-4.952e-05
beta2_ct	-50	10.04	50	-0.000101
beta2_ct	-50	9.775	50	-2.447e-05
beta2_ct	-50	9.871	50	-7.444e-05
L_omega2_z	-50	-3.671	50	-3.363e-05
L_epsilon2_z	-50	1.208	50	-4.873e-05
logkappa2	-6.313	-2.409	-0.4504	-0.0001644
logSigmaM	-50	0.5709	10	-0.002078

6.3 Diagnostics for encounter-probability component

Next, we check whether observed encounter frequencies for either low or high probability samples are within the 95% predictive interval for predicted encounter probability

```
Enc_prob = SpatialDeltaGLMM::Check_encounter_prob(Report = Report,
    Data_Geostat = Data_Geostat, DirName = DateFile)
```

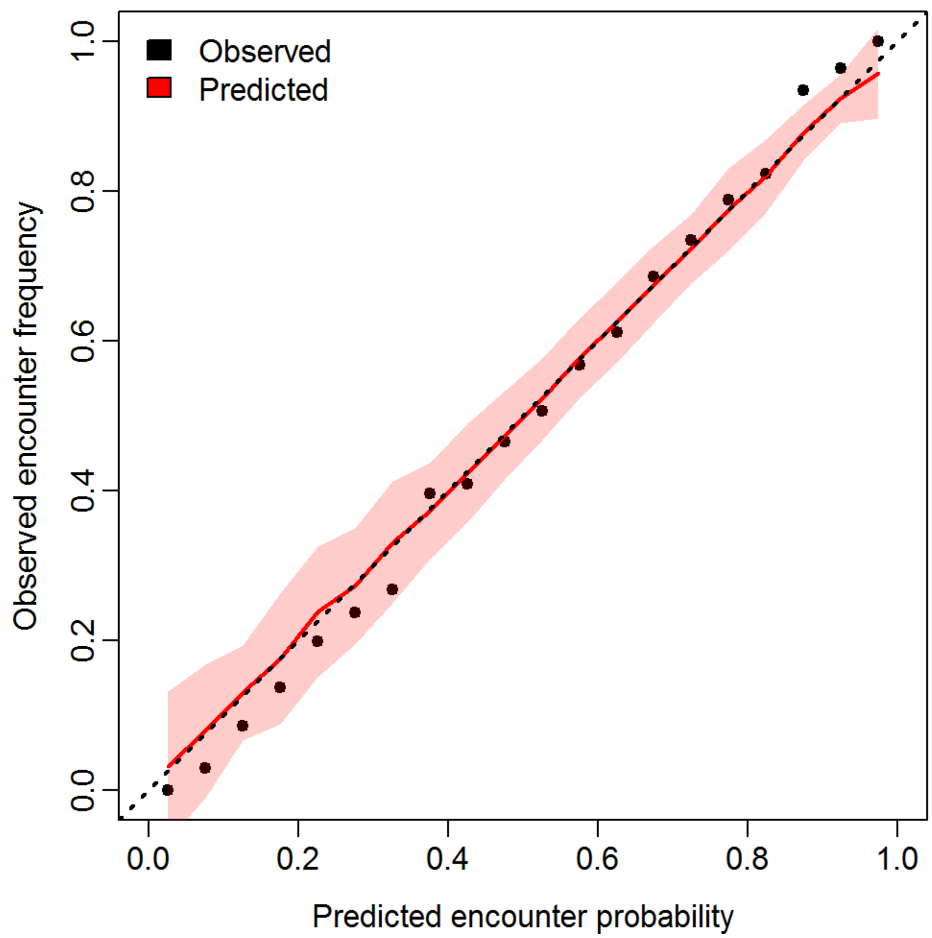

Figure 3: Expectated probability and observed frequency of encounter for “encounter probability” component

6.4 Diagnostics for positive-catch-rate component

We can visualize fit to residuals of catch-rates given encounters using a Q-Q plot. A good Q-Q plot will have residuals along the one-to-one line.

```
Q = SpatialDeltaGLMM::QQ_Fn(TmbData = TmbData, Report = Report,
    FileName_PP = paste0(DateFile, "Posterior_Predictive.jpg"),
    FileName_Phist = paste0(DateFile, "Posterior_Predictive-Histogram.jpg"),
    FileName_QQ = paste0(DateFile, "Q-Q_plot.jpg"),
    FileName_Qhist = paste0(DateFile, "Q-Q_hist.jpg"))  # SpatialDeltaGLMM::
```

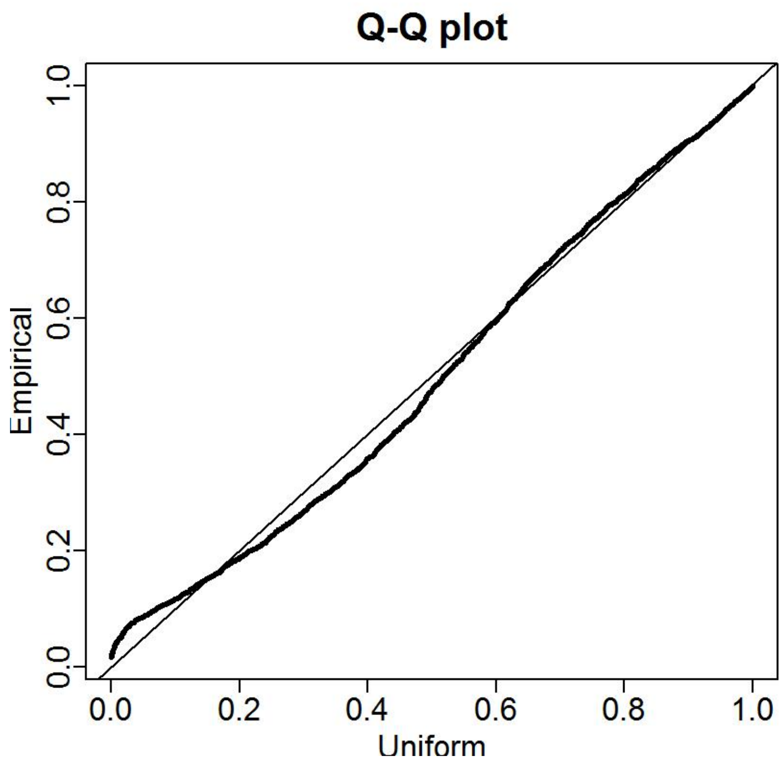

Figure 4: Quantile-quantile plot indicating residuals for "positive catch rate" component

6.5 Diagnostics for plotting residuals on a map

Finally, we visualize residuals on a map. To do so, we first define years to plot and generate plotting inputs. useful plots by first determining which years to plot (`Years2Include`), and labels for each plotted year (`Year_Set`)

```
# Get region-specific settings for plots
MapDetails_List = SpatialDeltaGLMM::MapDetails_Fn( "Region"=Region, "NN_Extrap"=Spatial_List$PolygonLis
# Decide which years to plot
Year_Set = seq(min(Data_Geostat[,'Year']),max(Data_Geostat[,'Year']))
Years2Include = which( Year_Set %in% sort(unique(Data_Geostat[,'Year'])))
```

We then plot Pearson residuals. If there are visible patterns (areas with consistently positive or negative residuals accross or within years) then this is an indication of the model "overshrinking" results towards the intercept, and model results should then be treated with caution.

```
SpatialDeltaGLMM::plot_residuals(Lat_i = Data_Geostat[,
    "Lat"], Lon_i = Data_Geostat[, "Lon"], TmbData = TmbData,
    Report = Report, Q = Q, savedir = DateFile, MappingDetails = MapDetails_List[["MappingDetails"]],
    PlotDF = MapDetails_List[["PlotDF"]], MapSizeRatio = MapDetails_List[["MapSizeRatio"]],
    Xlim = MapDetails_List[["Xlim"]], Ylim = MapDetails_List[["Ylim"]],
    FileName = DateFile, Year_Set = Year_Set, Years2Include = Years2Include,
    Rotate = MapDetails_List[["Rotate"]], Cex = MapDetails_List[["Cex"]],
    Legend = MapDetails_List[["Legend"]], zone = MapDetails_List[["Zone"]],
    mar = c(0, 0, 2, 0), oma = c(3.5, 3.5, 0, 0), cex = 1.8)
```

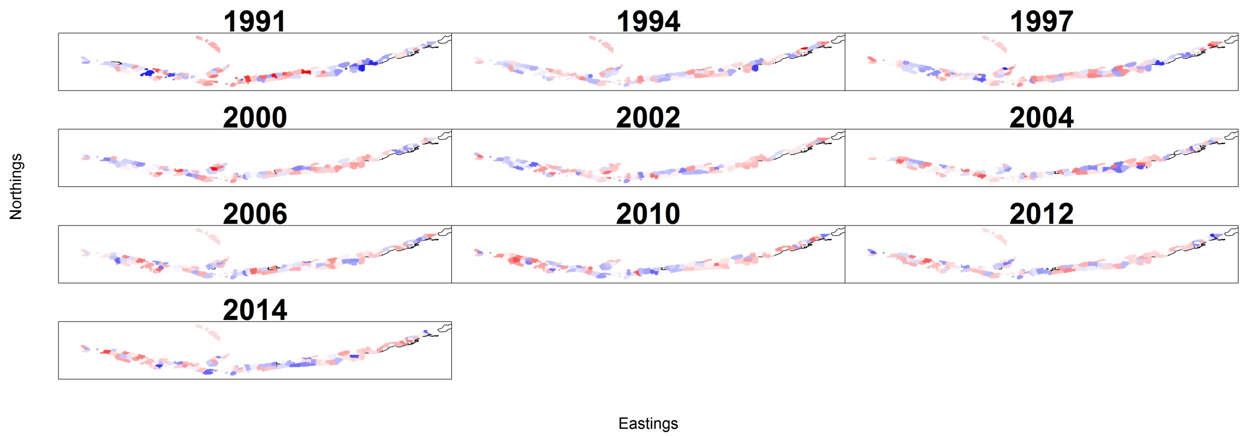

Figure 5: Pearson residuals for encounter-probability by knot

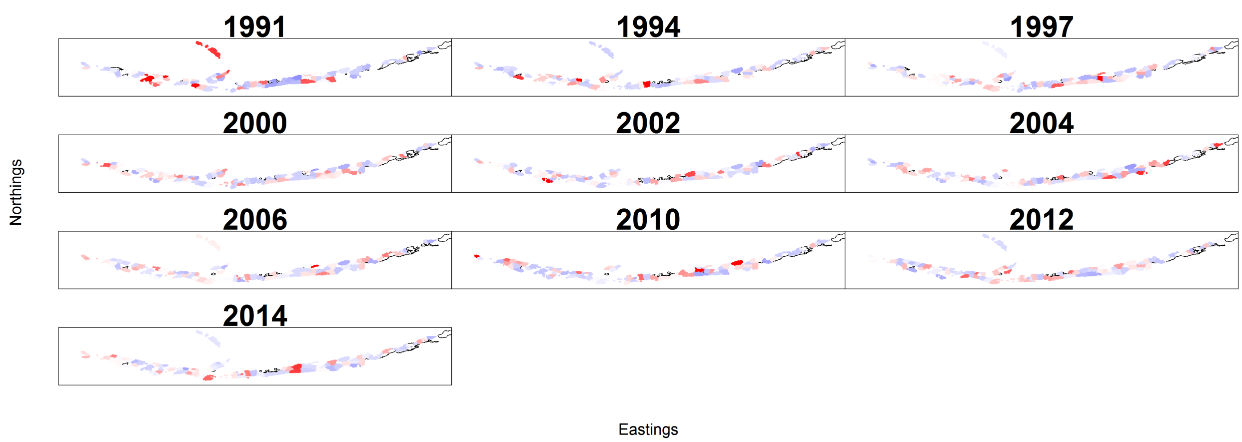

Figure 6: Pearson residuals for positive catch rates by knot

6.6 Model selection

To select among models, we recommend using the Akaike Information Criterion, AIC, via `Opt$AIC= 5.082\times 10^{4}`.

7 Model output

Last but not least, we generate pre-defined plots for visualizing results

7.1 Direction of "geometric anisotropy"

We can visualize which direction has faster or slower decorrelation (termed "geometric anisotropy")

```
SpatialDeltaGLMM::PlotAniso_Fn(FileName = paste0(DateFile,
    "Aniso.png"), Report = Report, TmbData = TmbData)
```

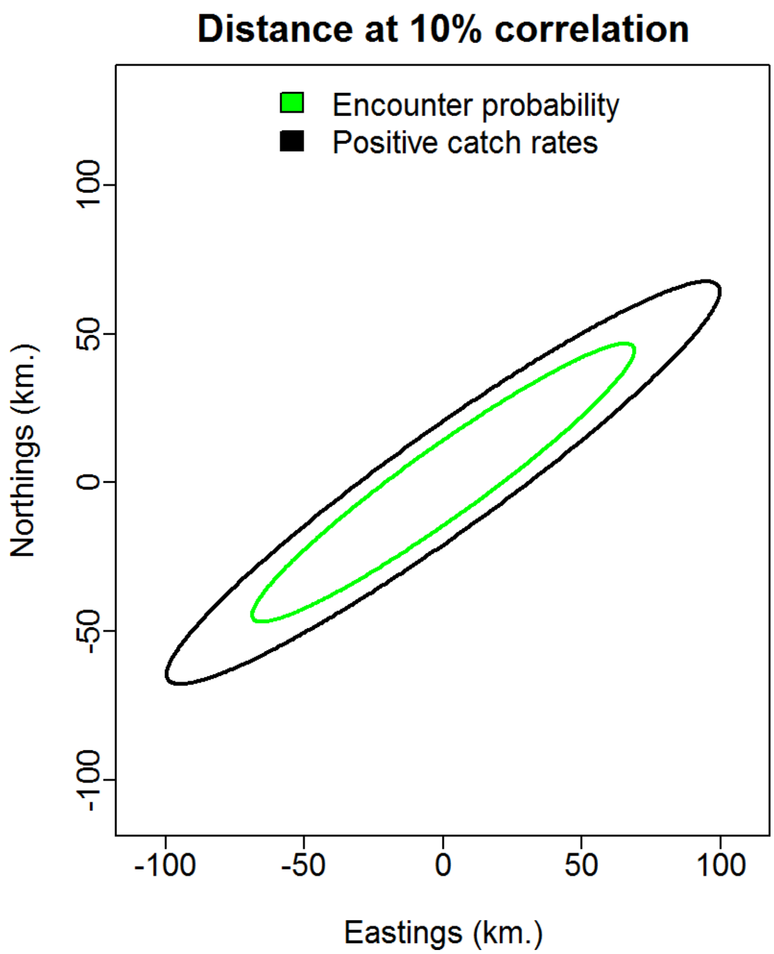

Figure 7: Decorrelation distance for different directions

7.2 Density surface for each year

We can visualize many types of output from the model. Here I only show predicted density, but other options are obtained via other integers passed to `plot_set` as described in `?PlotResultsOnMap_Fn`

```r
SpatialDeltaGLMM::PlotResultsOnMap_Fn(plot_set = c(3),
    MappingDetails = MapDetails_List[["MappingDetails"]],
    Report = Report, Sdreport = Opt$SD, PlotDF = MapDetails_List[["PlotDF"]],
    MapSizeRatio = MapDetails_List[["MapSizeRatio"]],
    Xlim = MapDetails_List[["Xlim"]], Ylim = MapDetails_List[["Ylim"]],
    FileName = DateFile, Year_Set = Year_Set, Years2Include = Years2Include,
    Rotate = MapDetails_List[["Rotate"]], Cex = MapDetails_List[["Cex"]],
    Legend = MapDetails_List[["Legend"]], zone = MapDetails_List[["Zone"]],
    mar = c(0, 0, 2, 0), oma = c(3.5, 3.5, 0, 0), cex = 1.8,
    plot_legend_fig = FALSE)
```

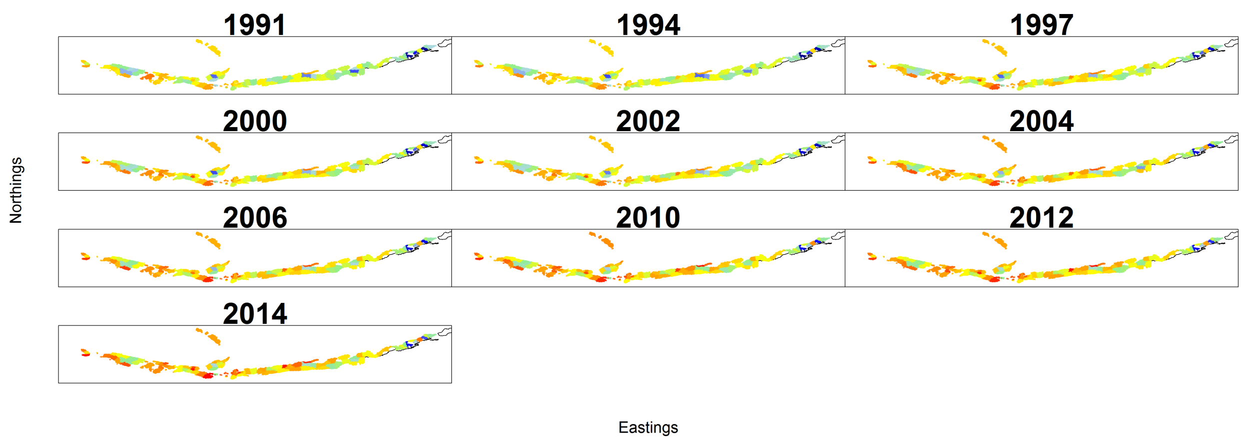

Figure 8: Density maps for each year

7.3 Index of abundance

The index of abundance is generally most useful for stock assessment models.

```r
Index = SpatialDeltaGLMM::PlotIndex_Fn(DirName = DateFile,
    TmbData = TmbData, Sdreport = Opt[["SD"]], Year_Set = Year_Set,
    Years2Include = Years2Include, use_biascorr = TRUE)
pander::pandoc.table(Index$Table[, c("Year", "Fleet",
    "Estimate_metric_tons", "SD_log", "SD_mt")])
```

Year	Fleet	Estimate_metric_tons	SD_log	SD_mt
1991	1	368960	0.1854	68410
1992	1	22587	0.3308	7471
1993	1	22587	0.3308	7471
1994	1	401435	0.1818	72981
1995	1	22587	0.3308	7471
1996	1	22587	0.3308	7471
1997	1	709689	0.1582	112241
1998	1	22587	0.3308	7471
1999	1	22587	0.3308	7471
2000	1	721955	0.1526	110178
2001	1	22587	0.3308	7471

Year	Fleet	Estimate_metric_tons	SD_log	SD_mt
2002	1	646127	0.1564	101066
2003	1	22587	0.3308	7471
2004	1	1037046	0.1585	164366
2005	1	22587	0.3308	7471
2006	1	1068484	0.1654	176736
2007	1	22587	0.3308	7471
2008	1	22587	0.3308	7471
2009	1	22587	0.3308	7471
2010	1	1420802	0.1596	226804
2011	1	22587	0.3308	7471
2012	1	1416894	0.144	204012
2013	1	22587	0.3308	7471
2014	1	1666206	0.14	233202

7.4 Center of gravity and range expansion/contraction

We can detect shifts in distribution or range expansion/contraction.

```
SpatialDeltaGLMM::Plot_range_shifts(Report = Report,
    TmbData = TmbData, Sdreport = Opt[["SD"]], Znames = colnames(TmbData$Z_xm),
    PlotDir = DateFile, Year_Set = Year_Set)
```

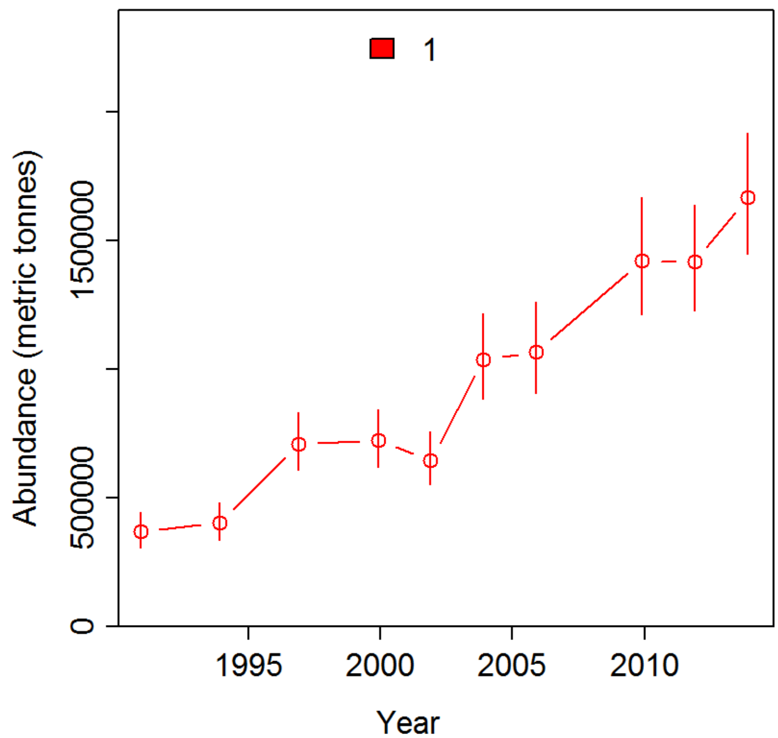

Figure 9: Index of abundance plus/minus 1 standard error

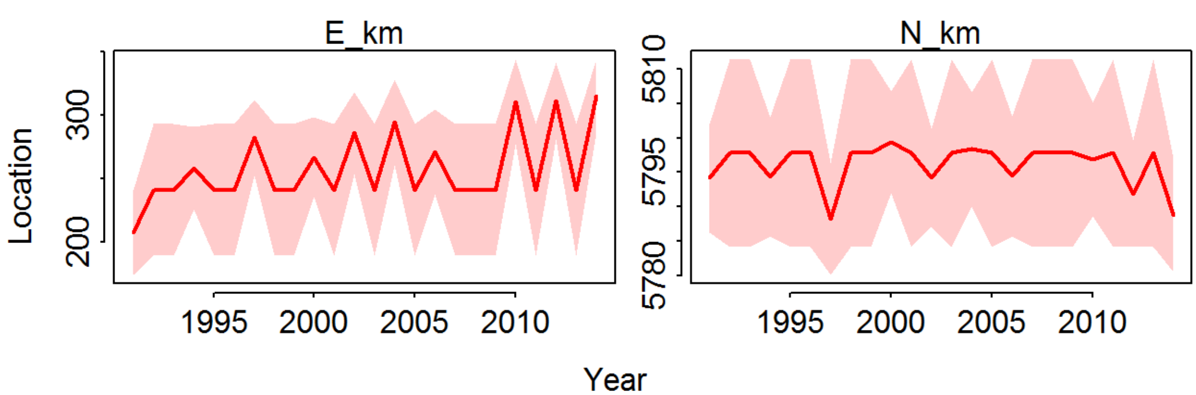

Figure 10: Center of gravity (COG) indicating shifts in distribution plus/minus 1 standard error

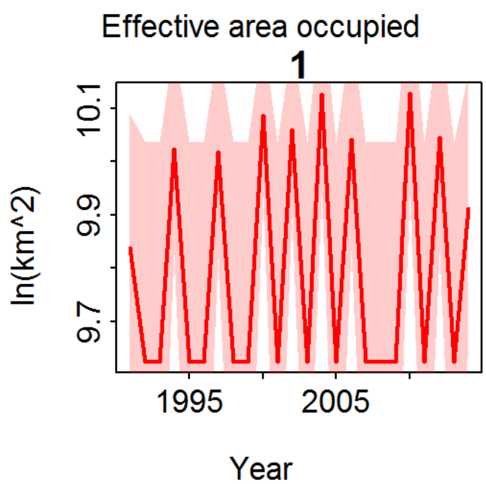

Figure 11: Effective area occupied indicating range expansion/contraction plus/minus 1 standard error