## Empirical orthogonal functions

James T. Thorson

```
library(tinyVAST)
library(fmesher)
set.seed(101)
```

tinyVAST is an R package for fitting vector autoregressive spatio-temporal (VAST) models. We here explore the capacity to specify a generalized linear latent variable model that is configured to generalize an empirical orthogonal function analysis.

## Empirical Orthogonal Function (EOF) analysis

To start, we reformat data on September Sea ice concentrations:

```
data( sea_ice )
library(sf)
library(rnaturalearth)
# project data
sf_ice = st_as_sf( sea_ice, coords = c("lon","lat") )
st_crs(sf_ice) = "+proj=longlat +datum=WGS84"
sf_ice = st_transform( sf_ice,
                      crs=st crs("+proj=laea +lat 0=90 +lon 0=-30 +units=km") )
sf_pole = st_point(c(0,90))
sf_pole = st_sfc( sf_pole, crs="+proj=longlat +datum=WGS84" )
sf_pole = st_transform( sf_pole, crs=st_crs(sf_ice) )
sf_pole = st_buffer( sf_pole, dist=3000 )
sf_ice = st_intersection( sf_ice, sf_pole )
#> Warning: attribute variables are assumed to be spatially constant throughout all geometries
Data = data.frame( st_drop_geometry(sf_ice),
              st_coordinates(sf_ice),
              var = "Ice" )
```

Next, we construct the various inputs to tiny VAST

Finally, we can extract, rotate, and plot the dominant modes of variability and associated spatial responses:

```
# Country shapefiles for plotting
sf maps = ne countries( return="sf", scale="medium", continent=c("north america", "europe", "asia") )
sf_maps = st_transform( sf_maps, crs=st_crs(sf_ice) )
sf_maps = st_union( sf_maps )
# Shapefile for water
sf water = st difference( st as sfc(st bbox(sf maps)), sf maps )
# Create extrapolation grid
cellsize = 50
sf_grid = st_make_grid( sf_pole, cellsize=cellsize )
# Restrict to water
grid_i = st_intersects( sf_water, sf_grid )
sf_grid = sf_grid[ unique(unlist(grid_i)) ]
# Restrict to 3000 km from North Pole
grid_i = st_intersects( sf_pole, sf_grid )
sf_grid = sf_grid[ unique(unlist(grid_i)) ]
newdata = data.frame( st_coordinates(st_centroid(sf_grid)),
                      var = "Ice" )
# Extract loadings
L_tf = matrix( 0, nrow=length(unique(Data$year)), ncol=2,
               dimnames=list(unique(Data$year),c("EOF 1","EOF 2")) )
L_tf[lower.tri(L_tf,diag=TRUE)] = out$opt$par[names(out$opt$par)=="beta_z"]
# Extract factor-responses
EOF1_g = predict( out, cbind(newdata, year="EOF_1"), what="pepsilon_g" )
EOF2_g = predict( out, cbind(newdata, year="EOF_2"), what="pepsilon_g" )
omega_g = predict( out, cbind(newdata, year="EOF_2"), what="pomega_g" )
# Rotate responses and loadings
rotated_results = rotate_pca( L_tf=L_tf, x_sf=cbind(EOF1_g,EOF2_g), order="decreasing" )
#> Warning in sqrt(Eigen$values): NaNs produced
EOF1 g = rotated results$x sf[,1]
E0F2_g = rotated_results$x_sf[,2]
```

