

Analysis steps Zandmotor benthos - density analysis

Peter M.J. Herman

May 2021

Introduction

This document contains the code and data for the different steps in analysis of the benthos data from the Zandmotor. We refer to the published paper (Herman et al., 2021, Frontiers in Marine Science) for all details.

The code starts with some general constants. A function to produce a compound png image, based on images from the separate years, is also given here as it is needed at several places in the code.

Packages and general constants

```
rm(list=ls())
require(reshape2)
require(raster)
require(gstat)
library(rgdal)
require(rgeos)
require(maps)
require(maptools)
require(mapdata)
require(vegan)
require(png)

##### projection strings #####
pro<-CRS(SRS_string='EPSG:4326')
proRD<-CRS(SRS_string='EPSG:28992')
par(mfrow=c(3,1))

# function to write png compound images
pngbasedir<-"./output/pngs/dens/"

compplot<-function(nam){
  xmin<-120
  xmax<-853
  ymin<-131
  ymax<-820
  nx<-xmax-xmin+1
  ny<-ymax-ymin+1
  tb<-array(data=0,dim=c(nx,ny+4*350+200,3))
  nf<-0
  for(jr in c(2010,2012,2013,2015,2017)){
    fn<-paste(pngbasedir,nam,"_",jr,".png",sep="")
    a<-readPNG(fn)
    a<-a[xmin:xmax,ymin:ymax,]
    a[a[,1]==1&a[,2]==1&a[,3]==1]<-0
```

```

    tb[, (nf*350+1):(nf*350+690),] <- tb[, (nf*350+1):(nf*350+690),] + a
    nf <- nf + 1
  }
  tb[tb[, , 1] == 0 & tb[, , 2] == 0 & tb[, , 3] == 0] <- -1
  a <- readPNG(fn)
  tb[, (ny+4*350+51):(ny+4*350+200),] <- a[xmin:xmax, 851:1000,]
  writePNG(tb, paste(pngbasedir, "all_", nam, ".png", sep = ""))
}

```

Reading in the benthos data

The benthic data were made available by WMR (Jeroen Wijsman) as binary files and a script to process and further structure these data. The script has been modified slightly to yield the basic files for the benthic analysis.

A subsection has been added to read in the sediment data and link them to the benthos data, in the station descriptions.

```

#####
#####
# This code has been run once, and the results stored as binary files
#####
#####

datadir <- "./input/benthosdata/"

#####
# Sampled Stations
AllStations <- readRDS(paste(datadir, "AllStations.rds", sep = ""))

# Only stations sampled with a Van Veen are selected here

Stations <- AllStations[AllStations$Van_Veen == 1,]

# Stations are coupled to data through the field StationNr
#####
# Benthos Data
AllData2 <- readRDS(paste(datadir, "AllData_Uncorr.rds", sep = ""))

# function to perform the swap of species names
swnames <- function(name_old, name_new, DF) {
  DF$NameAcc <- ifelse(DF$NameAcc == name_old, name_new, DF$NameAcc)
  return(DF)
}

# make sure all names are characters
AllData2$NameAcc <- AllData2$LATIJNLANG
AllData2$NameAcc <- as.character(AllData2$NameAcc)

# Correct species - initial step

AllData2 <- swnames("Abra", "Abra alba", AllData2)
AllData2 <- swnames("Aphelochaeta marioni", "Cirratulidae", AllData2)
AllData2 <- swnames("Capitella capitata", "Capitella", AllData2)
AllData2 <- swnames("Capitellidae", "Capitella", AllData2)
AllData2 <- swnames("Echinocardium", "Echinocardium cordatum", AllData2)

```

```

AllData2<-swnames("Ensis leei","Ensis",AllData2)
AllData2<-swnames("Eteone longa","Eteone",AllData2)
AllData2<-swnames("Eteone flava","Eteone",AllData2)
AllData2<-swnames("Eteoninae","Eteone",AllData2)
AllData2<-swnames("Euspira","Euspira nitida",AllData2)
AllData2<-swnames("Fabulina","Fabulina fabula",AllData2)
AllData2<-swnames("Grania","Grania postclitellochaeta",AllData2)
AllData2<-swnames("Harmothoe imbricata","Harmothoe",AllData2)
AllData2<-swnames("Harmothoe impar","Harmothoe",AllData2)
AllData2<-swnames("Jaera","Jaera (Jaera) albifrons",AllData2)
AllData2<-swnames("Lanice","Lanice conchilega",AllData2)
AllData2<-swnames("Liocarcinus","Liocarcinus holsatus",AllData2)
AllData2<-swnames("Lutraria","Lutraria lutraria",AllData2)
AllData2<-swnames("Malmgreniella","Malmgrenia",AllData2)
AllData2<-swnames("Marenzelleria","Marenzelleria viridis",AllData2)
AllData2<-swnames("Melita","Melitidae",AllData2)
AllData2<-swnames("Microprotopus","Microprotopus maculatus",AllData2)
AllData2<-swnames("Owenia fusiformis","Owenia",AllData2)
AllData2<-swnames("Pectinaria","Pectinariidae",AllData2)
AllData2<-swnames("Polynoinae","Polynoidae",AllData2)
AllData2<-swnames("Processa parva","Processa modica modica",AllData2)
AllData2<-swnames("Pseudocuma (Pseudocuma) simile","Pseudocuma simile",AllData2)
AllData2<-swnames("Scolelepis","Scolelepis (Scolelepis) squamata",AllData2)
AllData2<-swnames("Scoloplos","Scoloplos armiger",AllData2)
AllData2<-swnames("Sipunculus nudus","Sipuncula",AllData2)
AllData2<-swnames("Spio filicornis","Spio",AllData2)
AllData2<-swnames("Stenothoe","Stenothoe marina",AllData2)
AllData2<-swnames("Streblospio","Streblospio benedicti",AllData2)
AllData2<-swnames("Streblospio shrubsolii","Streblospio benedicti",AllData2)
AllData2<-swnames("Tellina","Tellinidae",AllData2)
AllData2<-swnames("Tharyx","Cirratulidae",AllData2)
AllData2<-swnames("Urothoe","Urothoe poseidonis",AllData2)
AllData2<-swnames("Venerupis","Venerupis corrugata",AllData2)
AllData2<-swnames("Grania postclitellochaeta","Oligochaeta",AllData2)
AllData2<-swnames("Naididae","Oligochaeta",AllData2)
AllData2<-swnames("Tubificidae","Oligochaeta",AllData2)
AllData2<-swnames("Tubificoides benedii","Oligochaeta",AllData2)
AllData2<-swnames("Tubificoides brownae","Oligochaeta",AllData2)
AllData2<-swnames("Tubificoides diazi","Oligochaeta",AllData2)
AllData2<-swnames("Alcyonidium parasiticum","Bryozoa",AllData2)
AllData2<-swnames("Conopeum reticulum","Bryozoa",AllData2)
AllData2<-swnames("Electra pilosa","Bryozoa",AllData2)
AllData2<-swnames("Actinaria","Anthozoa",AllData2)
AllData2<-swnames("Anthoathecata","Hydrozoa",AllData2)
AllData2<-swnames("Campanulariidae","Hydrozoa",AllData2)
AllData2<-swnames("Clytia hemisphaerica","Hydrozoa",AllData2)
AllData2<-swnames("Leptothecata","Hydrozoa",AllData2)
AllData2<-swnames("Obelia bidentata","Hydrozoa",AllData2)
AllData2<-swnames("Obelia longissima","Hydrozoa",AllData2)
AllData2<-swnames("Tubulariidae","Hydrozoa",AllData2)
AllData2<-swnames("Tubulanus polymorphus","Nemertea",AllData2)
AllData2<-swnames("Anaitides","Phyllodoce",AllData2)
AllData2<-swnames("Angulus fabula","Fabulina fabula",AllData2)

```

```

AllData2<-swnames("Angulus tenuis","Macomangulus tenuis",AllData2)
AllData2<-swnames("Dendrobrachiata/Caridea","Decapoda",AllData2)
AllData2<-swnames("Malmgreniella darbouxi","Malmgrenia darbouxi",AllData2)
AllData2<-swnames("Malmgreniella lunulata","Malmgrenia darbouxi",AllData2)
AllData2<-swnames("Mysella bidentata","Kurtiella bidentata",AllData2)
AllData2<-swnames("Nassarius nitidus","Nassarius",AllData2)
AllData2<-swnames("Nassarius reticulatus","Nassarius",AllData2)
AllData2<-swnames("Ophiurida","Ophiuroidea",AllData2)
AllData2<-swnames("Pectinaria koreni","Pectinariidae",AllData2)
AllData2<-swnames("Processa modica","Processa modica modica",AllData2)
AllData2<-swnames("Scolelepis squamata","Scolelepis (Scolelepis) squamata",AllData2)
AllData2<-swnames("Scoloplos (Scoloplos) armiger","Scoloplos armiger",AllData2)

```

remove some groups

```

AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Animalia"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Insecta"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Scatella"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Ephydriidae"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Xanthocanace ranula"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Ammodytes"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Ammodytes marinus"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Ammodytes tobianus"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Overig"))
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Pisces"))

```

Further corrections after analysis of differences between labs

```

AllData2<-swnames("Spisula","Spisula subtruncata",AllData2)
AllData2<-swnames("Diastylis","Diastylis bradyi",AllData2)
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Brachyura"))
AllData2<-swnames("Harmothoe","Malmgrenia",AllData2)
AllData2<-swnames("Malmgrenia lunulata","Malmgrenia darbouxi",AllData2)
AllData2<-swnames("Nephtys kersivalensis","Nephtys cirrosa",AllData2)
AllData2<-swnames("Pariambus typicus","Caprellidae",AllData2)
AllData2<-swnames("Eumida bahusiensis","Eumida",AllData2)
AllData2<-swnames("Eumida sanguinea","Eumida",AllData2)
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Chaetognatha"))
AllData2<-swnames("Glycera tridactyla","Glycera",AllData2)
AllData2<-swnames("Glycera alba","Glycera",AllData2)
AllData2<-swnames("Spio goniocephala","Spio",AllData2)
AllData2<-swnames("Spio decorata","Spio",AllData2)
AllData2<-swnames("Spio symphyta","Spio",AllData2)
AllData2<-swnames("Spio martinensis","Spio",AllData2)
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Terebellida"))
AllData2<-swnames("Ophelia borealis","Ophelia",AllData2)
AllData2<-swnames("Ophelia limacina","Ophelia",AllData2)
AllData2 <-swnames("Phyllodoce","Phyllodocidae",AllData2)
AllData2<-swnames("Spionida","Spio",AllData2)
AllData2<-swnames("Spionidae","Spio",AllData2)
AllData2<-swnames("Ophiura","Ophiuroidea",AllData2)

```

```

AllData2<-swnames("Ophiuridae","Ophiuroidea",AllData2)
AllData2<-swnames("Nereidinae","Nereididae",AllData2)
AllData2 <- droplevels(subset(AllData2,AllData2$NameAcc!="Nematoda"))
AllData2<-swnames("Mytilidae","Mytilus edulis",AllData2)
AllData2<-swnames("Mysidae","Mysida",AllData2)
AllData2<-swnames("Mactridae","Mactra stultorum",AllData2)
AllData2<-swnames("Gammarus","Gammaridea",AllData2)
AllData2<-swnames("Crangonidae","Crangon crangon",AllData2)
AllData2<-swnames("Corophiidae","Corophium",AllData2)
AllData2<-swnames("Balanidae","Cirripectida",AllData2)

# Correct for subsampling
AllData2$AANTAL <- AllData2$WAARDE*AllData2$SUBFACTOR
AllData2$AFDW <- AllData2$AFDW*AllData2$SUBFACTOR
AllData2 <- AllData2[AllData2$NameAcc!="",]

# Calculate species density and biomass per station
SpecStationAantal <- dcast(StationNr~NameAcc,data=AllData2,
                           fun.aggregate=function(x)sum(x,na.rm=T),
                           value.var="AANTAL")
names(SpecStationAantal)[2:ncol(SpecStationAantal)]<-
  paste("d_",names(SpecStationAantal)[2:ncol(SpecStationAantal)],sep="")
SpecStationAFDW <- dcast(StationNr~NameAcc,data=AllData2,
                        fun.aggregate=function(x)sum(x,na.rm=T),
                        value.var="AFDW")
names(SpecStationAFDW)[2:ncol(SpecStationAFDW)]<-
  paste("b_",names(SpecStationAFDW)[2:ncol(SpecStationAFDW)],sep="")
SpecStationAFDW<-SpecStationAFDW[, -1]
SpecStat<-cbind(SpecStationAantal,SpecStationAFDW)

# couple with AllStations
AllData <- merge(Stations,SpecStat,by = "StationNr",all=TRUE)
# select van Veen only
advv<-AllData[AllData$TYPE=="Van Veenhapper",]
# remove incomplete year 2011
advv<-advv[advv$Jaar!=2011,]
# remove year 2019
advv<-advv[advv$Jaar!=2019,]
# remove NA station
advv <- advv[!is.na(advv$StationNr),]
# split in station description and community matrix
stats<-advv[,1:25]
vals<-advv[,26:ncol(advv)]
# calculate density based on observed value and sampled surface
for (i in 1:ncol(vals)){
  vals[,i]<-vals[,i]/stats$HAPGROOTTE
  vals[,i][is.na(vals[,i])]<-0
}
# split in density and biomass files
dens<-vals[,1:(ncol(vals)/2)]
biom<-vals[, (ncol(vals)/2+1):ncol(vals)]
for (i in 1:length(names(dens)))names(dens)[i]<-gsub("d_","",names(dens)[i])
for (i in 1:length(names(biom)))names(biom)[i]<-gsub("b_","",names(biom)[i])

```

```

#
# some species have 0 throughout for density or biomass. We drop them. They
# are in the list because they occurred in stations that we dropped
ll<-which(apply(dens,2,sum)==0 | apply(biom,2,sum)==0)
dens<-dens[,-ll]
biom<-biom[,-ll]

#####

# read in sediment data

zms<-read.csv("./input/sedimentdata/SedimentZM2010_2017.csv",stringsAsFactors=F)
zms<-zms[,-c(1,29:34)]
names(zms)[1]<-"StationNr"
stats<-merge(stats,zms,by="StationNr",all.x=T)
stats$StationNr<-as.character(stats$StationNr)
stats$X_RD<-as.numeric(stats$X_RD)
stats$Y_RD<-as.numeric(stats$Y_RD)
stats$TRANSECT<-as.numeric(as.character(stats$TRANSECT))
stats$DiepteKlas<-as.factor(stats$DiepteKlas)
stats$HoogteKlas<-as.factor(stats$HoogteKlas)
stats$Remarks<-as.character(stats$Remarks)
stats$depth<- -stats$depth
# remove unnecessary variables
weg<-c(2,3,12,13,14,15,17,19,22,23,26,27,28,29,30,31,44,45,46,47,48,50)
#      "ID", "TYPE", "Van_Veen",
#      "Schaaf", "Sediment", "OrgC", "Datum", "Remarks", "PROJECTNAAM",
#      "APPARAAT", "datestr", "lat", "lon", "x", "y", "depth",
#      "SSILT16"      "SSILT2"      "SSILT32"      "SSILT4"      "SSILT50",
#      "SSILT8"
stats<-stats[,-weg]
# give names that are easy to use
names(stats)<-c("StationNr","transect","x_rd","y_rd","lat","lon","dep_cm",
               "dep_cl","hei_cl","lab","dateschaaf","date","jaar",
               "sampsurf","sampunit","totn","cnratio",
               "scourse","sd10um","sd90um","sd50um","sd50phi","sfines",
               "smedium","orgc","spsa","ssd",
               "ssilt63","svfines")
# interpolate sediment data of 2010_43, in order to avoid NAs
stats[22,16:29]<-(stats[21,16:29]+stats[23,16:29])/2

#####
save(stats,dens,biom,file="./output/binary_files/commatsvv.Rdata")
#reorganize stats for output
stats<-stats[,c(1,2,3,4,5,6,7,10,12,13,14,15,16,17,25,28,29,23,24,18,19,21,20,22,26,27)]
write.csv(stats,file="./output/csv_output/grainsize.csv",row.names=F)
write.csv(dens,file="./output/csv_output/dens.csv",row.names = F)
write.csv(biom,file="./output/csv_output/biom.csv",row.names = F)
save(AllData,file="./output/binary_files/AllData.Rdata")

rm(advv,AllData2,AllStations,SpecStat,SpecStationAantal,SpecStationAFDW,
    Stations,vals,zms,swnames,ll)

```

Read in the model data

Arjen Luijendijk provided us with output from a hydrodynamic model, which was run for one month in each of the years for which benthos data are available. The files were provided as a set of shape and ascii files, from which rasters were derived for each of the years and the different variables. These rasters were stored together into a brick structure (package raster).

The model output consists of shapefiles, which give the geometry of the model cells as polygons, and *.dep files which contain the model output variables in a particular format. The *.dep files are ascii files with 12 columns and as many rows as needed to complete the grid. Its structure is rather complicated. Rather than finding this out in general, we took one file where we were sure that all meaningful values were given, and all other values (e.g. boundaries etc.) indicated as non-available. The pattern of non-availables in this file was then used to mask all other files, keeping only meaningful values in the right order. This could be checked because for some variables, an x-y-z file (taugrid.txt) was available. Therefore the below ad-hoc method works.

```
#####
#####
# This code has been run once, and the results stored as binary files
#####
#####

### function to restructure a raw table into a linear vector
rf<-function(rawtable){
  outvec<-vector(length=(nrow(rawtable)*12))
  t<-0
  for (row in 1:nrow(rawtable)){
    for (col in 1:12){
      t<-t+1
      outvec[t]<-rawtable[row,col]
    }
  }
  return(outvec)
}

# determine the template for the dep files and the coordinates of the cell centres
tf<-read.table("./input/modeldata/max_flow_wave_related_bed_shear_stress/2010.dep",fill=T,na.strings=c(
tfv<-rf(tf)
wegpattern<-which(is.na(tfv))
tg2012<-read.table("./input/modeldata/mean_flow_related_bed_shear_stress/2012_taugrid.txt")
xy<-data.frame(X=tg2012$V1,Y=tg2012$V2)
#### end template and xy coordinates

dirs<-c("bathymetry",
        "max_flow_velo",
        "max_flow_wave_related_bed_shear_stress",
        "mean_flow_related_bed_shear_stress",
        "mean_flow_velo")
prefix<-c("", "Max_flow_velo_", "", "", "Mean_flow_velo_")
years<-c("2010", "2011", "2012", "2013", "2014", "2015", "2017")
colprefix<-c("bath", "maxflow", "tauflow", "tautot", "meanflow")

vals<-xy
for(i in 1:length(dirs)){
  dir<-dirs[i]
```



```

for (year in years){
  filnam<-paste("./input/modeldata/",dir,"/",prefix[i],year,".dep",sep="")
  rawtable<-read.table(filnam,fill=T,na.strings=c("NA","-999.00000000"))
  ov<-rf(rawtable)
  ov<-ov[-wegpattern]
  outfil<-paste("./input/modeldata/",dir,"/",prefix[i],year,"vals.txt",sep="")
  write.table(ov,file=outfil,row.names=F,col.names=F)
  nams<-c(names(vals),paste(colprefix[i],"_",year,sep=""))
  vals<-cbind(vals,ov)
  names(vals)<-nams
}
}
save(vals,file="./output/binary_files/modeloutput.Rdata")
write.csv(vals,file="./output/csv_output/modeloutput.csv",row.names=F)
rm(rawtable,tf,tg2012,xy,colprefix,dir,dirs,filnam,nams,
  outfil,ov,prefix,tfv,wegpattern,year,years,rf)

```

linking observations to model output

One important link between benthos observations and model output is that we need to have the values of the model output variables in every station where also benthos data were collected. In order to do this, first the model output is interpolated (nearest-neighbour) into a high-resolution raster. All of these rasters are stored in a brick structure. Subsequently the value of this raster for each of the sampling points is read in, using the 'extract' function. As there are model fields for each of the years, care has to be taken that the output from the correct year is used. This is achieved by running a loop over years.

We also do the reverse for grain size. We have values at the sampling stations but not in between. A raster of grain size is produced by interpolating between the observation values.

```

#####
#####
# This code has been run once, and the results stored as binary files
#####
#####

#### function to interpolate a set of model results and store it into a raster
intvals<-function(rint,edf,nam){
  mg <- gstat(id = nam, formula = z~1, data=edf,nmax=4, set=list(idp = 1))
  rout<-interpolate(rint,mg)
  return(rout)
}
##### end interpolation function

load("./output/binary_files/modeloutput.Rdata") # creates the data frame vals

#### template raster
coordinates(vals)<- ~X+Y
proj4string(vals)<-proRD
r<-raster(nrows=350,ncols=300)
proj4string(r)<-proRD
extent(r)<- extent(vals)

#### create SpatialPolygons of model grid area
xy<-data.frame(X=coordinates(vals)[,1],Y=coordinates(vals)[,2])

```



```

clo<-xy[which(xy$Y==min(xy$Y)),]
cle<-xy[which(xy$X==min(xy$X)),]
cup<-xy[which(xy$Y==max(xy$Y)),]
cri<-xy[which(xy$X==max(xy$X)),]
po<-Polygon(rbind(clo,cle,cup,cri,clo),hole=F)
pos<-Polygons(list(po),"modelgrid")
mgrid<-SpatialPolygons(list(pos))
proj4string(mgrid)<-proRD
####
for(i in 1:ncol(vals)){
  edf<-vals[,i]
  names(edf)<-c("z")
  rout<-intvals(r,edf,names(vals)[i])
  names(rout)<-names(vals)[i]
  rout<-mask(rout,mgrid)
  if(i==1)b<-brick(rout) else b<-brick(rout,b)
}
##### Spatially interpolate Grain Size #####
for (jr in c(2010,2012,2013,2015,2017)){
  welk<- which(stats$jaar==jr)
  mgs <- stats[welk,]
  coordinates(mgs)<- ~x_rd+y_rd
  crs(mgs)<-proRD
  vzd <- variogram(sd50um~1, data=mgs,alpha=c(0,45))
  mzd <- fit.variogram(vzd, vgm(6000, "Exp", 1000,2000,anis=c(45,.5)))
  gzd <- gstat(NULL, "GS", sd50um~1, mgs, model=mzd)
  rmeanGS <- interpolate(r, gzd)
  rmeanGS <- mask(rmeanGS,mgrid)
  names(rmeanGS)<-paste("sd50um_",jr,sep="")
  b<-brick(rmeanGS,b)
}
##### END grain size #####
##### save brick #####
save(mgrid,file="./output/binary_files/mgrid.Rdata")
writeRaster(b,filename="./output/binary_files/ModelBrick.grd",overwrite=TRUE)

```

The following code reads in the brick of model results, and uses this to plot all the model output rasters, and also to estimate the values of the model output at the sampling points.

```

load("./output/binary_files/mgrid.Rdata")
b<-brick("./output/binary_files/modelbrick.grd")

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO"): Discarded
## datum Unknown based on Bessel 1841 ellipsoid in CRS definition

load("./output/binary_files/commatsvv.Rdata")

load("./output/binary_files/modeloutput.Rdata") # creates the data frame vals

#### template raster
coordinates(vals)<- ~X+Y
proj4string(vals)<-proRD
r<-raster(nrows=350,ncols=300,crs=proRD)
extent(r)<- extent(vals)
#

```

```
##### store essential stats variables in smpca #####
smpca<-stats[,c(3,4,7,13,16:29)]
smpca$lab<-as.factor(stats$lab)
##### make smpca spatial #####
coordinates(smpca)<- ~x_rd+y_rd
proj4string(smpca)<-proRD
##### find model output at sampling points #####
smpca$mnf<-smpca$tt<-smpca$tf<-smpca$mx<-smpca$depth<-
  smpca$depdif<-rep(NA,nrow(smpca))

for (jr in c(2010,2012,2013,2015,2017)){
  rmnf<-subset(b,paste("meanflow_",jr,sep=""))
  rtt <-subset(b,paste("tautot_",jr,sep=""))
  rtf <-subset(b,paste("tauflow_",jr,sep=""))
  rmxf<-subset(b,paste("maxflow_",jr,sep=""))
  rba <-subset(b,paste("bath_",jr,sep=""))
  if(jr==2010)lastrba<-rba
  welk<-which(smpca$jaar==jr)
  smpca$mnf[welk]<-extract(rmnf,smpca[welk,])
  smpca$tt[welk]<-extract(rtt,smpca[welk,])
  smpca$tf[welk]<-extract(rtf,smpca[welk,])
  smpca$mx[welk]<-extract(rmxf,smpca[welk,])
  smpca$depth[welk]<-extract(rba,smpca[welk,])
  depdif<- 0-rba+lastrba
  smpca$depdif[welk]<-extract(depdif,smpca[welk,])
  lastrba<-rba
}
```

Basic description of the benthic fauna

Here we list dominant species, in density and biomass, and perform a number of other basic analyses of the benthic fauna. The following code lists the dominant species and their share in the total.

```
totinds<-sum(dens)
nsamp<-nrow(dens)
nspec<-ncol(dens)
totbiom<-sum(biom)
specdens<-apply(dens,2,sum)
specbiom<-apply(biom,2,sum)
champion<-which(specdens==max(specdens))
fracensis<-specdens[champion]/totinds
fracensisbiom<-specbiom[champion]/totbiom
tt<-data.frame(ensis=dens[,45],jaar=as.factor(stats$jaar))
ensisjaardens<-aggregate(ensis~jaar,tt,FUN=mean)
tt<-data.frame(ensis=biom[,45],jaar=as.factor(stats$jaar))
ensisjaarbiom<-aggregate(ensis~jaar,tt,FUN=mean)
ed1<-mean(ensisjaardens[1:3,2])
ed2<-mean(ensisjaardens[4:5,2])
eb1<-mean(ensisjaarbiom[1:3,2])
eb2<-mean(ensisjaarbiom[4:5,2])
# dominant species in abundance
for (jr in c(2010,2012,2013,2015,2017)){
  jrdens<-apply(dens[stats$jaar==jr,],2,mean)
  jrdens<-jrdens[order(jrdens)]
}
```

```

domm<-paste(names(jrdens)[196:192], " (",
            round(jrdens[196:192],0),')',sep="")
if(jr==2010) domnm<-domm else domnm<-rbind(domnm,domm)
}
domnm<-t(domnm)
domnm<-as.data.frame(domnm)
names(domnm)<-c("2010","2012","2013","2015","2017")
rownames(domnm)<-c("1","2","3","4","5")
# dominant species in biomass
for (jr in c(2010,2012,2013,2015,2017)){
  jrbiom<-apply(biom[stats$jaar==jr,],2,mean)
  jrbiom<-jrbiom[order(jrbiom)]
  dommb<-paste(names(jrbiom)[196:192], " (",
              round(jrbiom[196:192],2),')',sep="")
  if(jr==2010) domnmb<-dommb else domnmb<-rbind(domnmb,dommb)
}
domnmb<-t(domnmb)
domnmb<-as.data.frame(domnmb)
names(domnmb)<-c("2010","2012","2013","2015","2017")
rownames(domnmb)<-c("1","2","3","4","5")

```

Here we plot species accumulation curves, rank abundance and rank biomass plots, to see how diversity and dominance have changed over the years

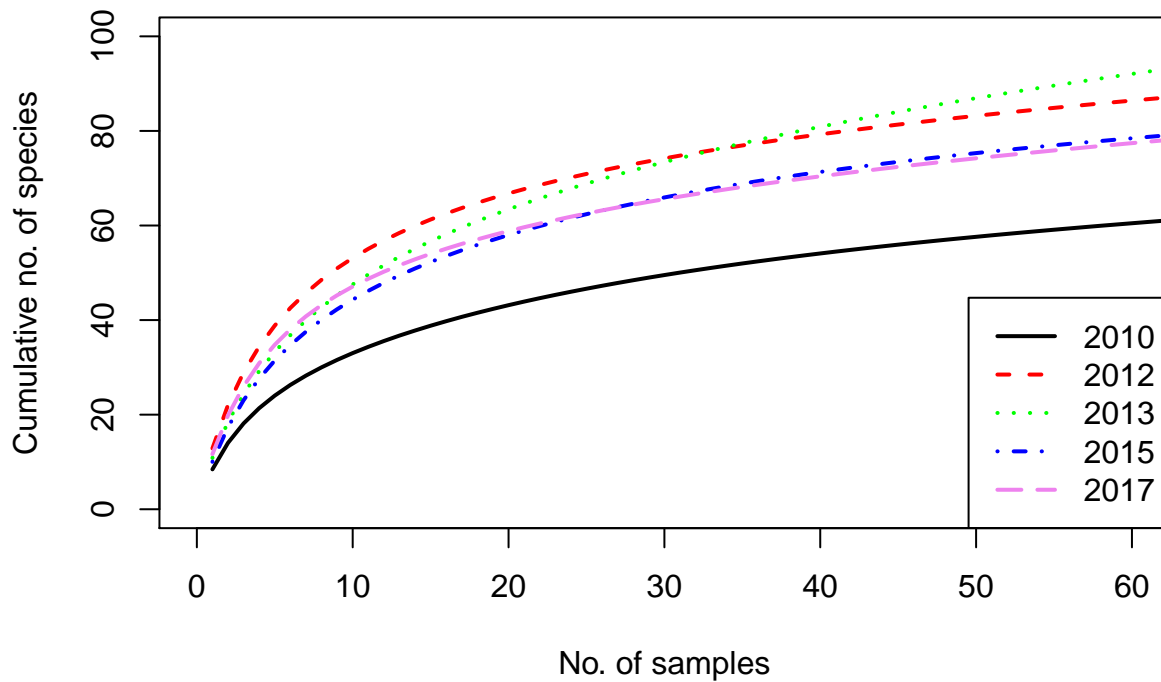
```

specnspec<-read.csv("./input/benthosdata/specnspec.csv",header=T,stringsAsFactors = FALSE)
cls<-c("black","red","green","blue","violet")
ncl<-0
for(jr in c(2010,2012,2013,2015,2017)){
  # uncomment the following two lines, and comment the line following them,
  # to test for the effect of having fewer samples in 2010
  # than in other years - there is hardly any effect at all

  smpls<-which(stats$jaar==jr)
  selsmp<-sample(smpls,62,replace = FALSE)
  # selsmp<-which(stats$jaar==jr)
  selspecs<-which(names(dens) %in% specnspec$Taxon_name[specnspec$valid==1])
  tt<-dens[selsmp,selspecs]
  sac<-specaccum(tt,method='exact')
  ncl<-ncl+1
  if(jr==2010)plot(sac,main="Species Accumulation curves",xlim=c(0,60),
                  ylim=c(0,100),col=cls[ncl],lty=ncl,lwd=2,ci=0,
                  xlab="No. of samples",
                  ylab="Cumulative no. of species") else
    plot(sac,add=T,col=cls[ncl],lty=ncl,lwd=2,ci=0)
}
legend("bottomright",legend=c("2010","2012","2013","2015","2017"),
      col=cls,lty=1:5,lwd=2)

```

Species Accumulation curves



```
#Rank abundance and Rank biomass
par(mfrow=c(1,2))

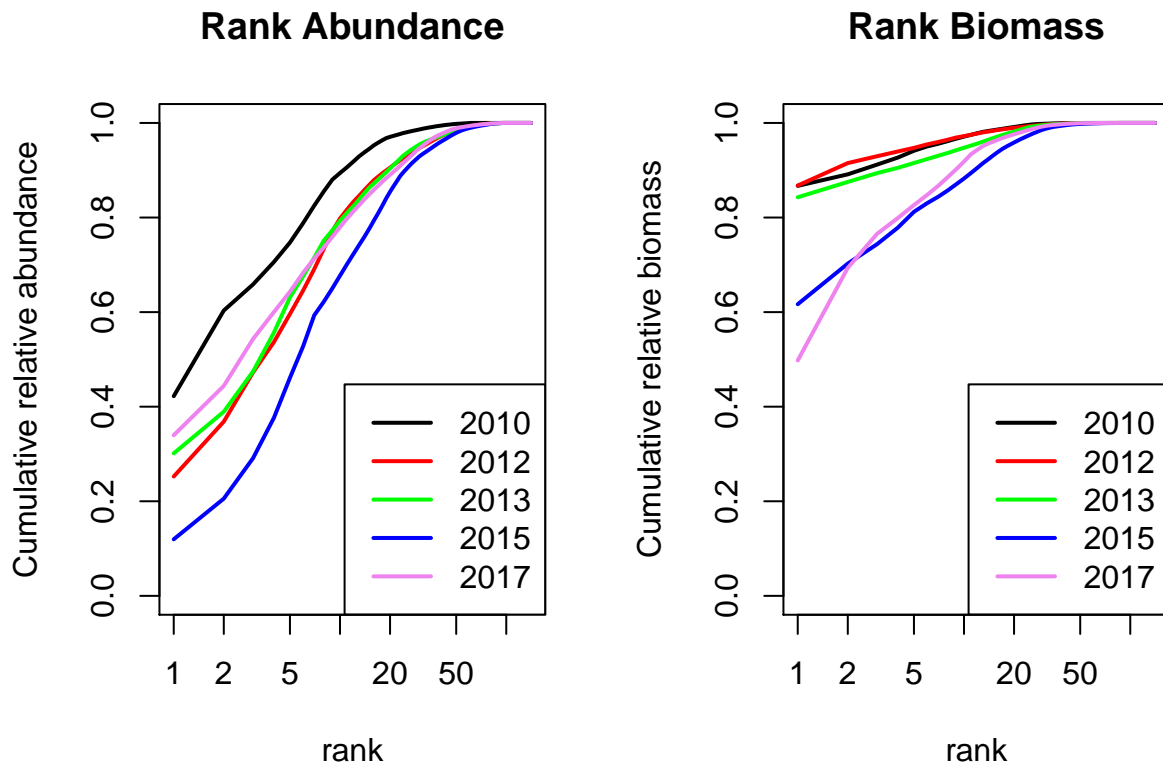
# rank abundance plots
ncl=0
for (jr in c(2010,2012,2013,2015,2017)){
  selsmp<-which(stats$jaar==jr)
  selspecs<-which(names(dens) %in% specnspec$Taxon_name[specnspec$valid==1])
  ttt<-dens[selsmp,selspecs]
  tt<-apply(ttt,2,sum)
  tt<-tt[order(tt,decreasing=T)]
  for(i in (2:length(tt)))tt[i]<-tt[i]+tt[i-1]
  tt<-tt/max(tt)
  ncl<-ncl+1
  if(jr==2010)plot(1:length(tt),tt,ylim=c(0,1),
                  log="x",col=cls[ncl],type="l",lwd=2,
                  main="Rank Abundance",
                  xlab="rank",
                  ylab="Cumulative relative abundance") else
    lines(1:length(tt),tt,col=cls[ncl],lwd=2)
}
legend("bottomright",legend=c("2010","2012","2013","2015","2017"),
      col=cls,lty=1,lwd=2)

# rank biomass plots
ncl=0
for (jr in c(2010,2012,2013,2015,2017)){
```

```

selsmp<-which(stats$jaar==jr)
selspecs<-which(names(biom) %in% specnspec$Taxon_name[specnspec$valid==1])
ttt<-biom[selsmp,selspecs]
tt<-apply(ttt,2,sum)
tt<-tt[order(tt,decreasing=T)]
for(i in (2:length(tt)))tt[i]<-tt[i]+tt[i-1]
tt<-tt/max(tt)
ncl<-ncl+1
if(jr==2010)plot(1:length(tt),tt,ylim=c(0,1),
                 log="x",col=cls[ncl],type="l",lwd=2,
                 main="Rank Biomass",
                 xlab="rank",
                 ylab="Cumulative relative biomass") else
  lines(1:length(tt),tt,col=cls[ncl],lwd=2)
}
legend("bottomright",legend=c("2010","2012","2013","2015","2017"),
      col=cls,lty=1,lwd=2)

```



```

par(mfrow=c(1,1))
#####

```

The following analysis calculates, for species that were absent in 2010 but found afterwards, the probability of not finding them in 2010 based on the zero term of a Poisson distribution.

```

# which species do not occur in 2010

```

```

tt<-dens[stats$jaar==2010,names(dens)%in% specnspec$Taxon_name[specnspec$valid==1]]

```

```

tttot<-apply(tt,2,sum)
spec0<-which(tttot==0)
frsp0<-apply(dens[stats$jaar!=2010,names(dens)%in% specnspec$Taxon_name[specnspec$valid==1]],2,
FUN=function(x)length(x[x>0]))
lambda<-frsp0[spec0]/nrow(dens[stats$jaar!=2010,])*62
critsp<-exp(-lambda)[exp(-lambda)<0.05/length(spec0)]
print(critsp)

```

```

##      Caprellidae Donax vittatus
##      1.941177e-04    1.282448e-06

```

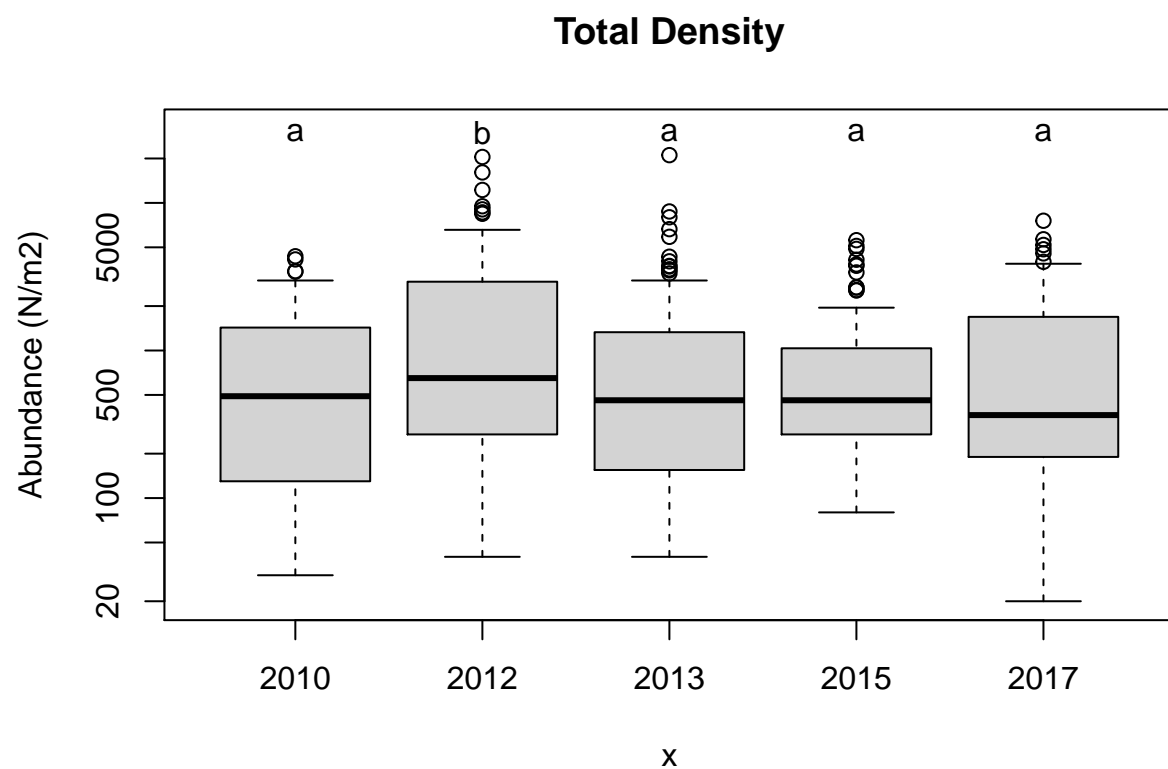
Average individual size

Here we investigate whether colonisation by small individuals was happening after the construction of the sand motor. We start the exploration by plotting average individual size (across all species) to see whether there is any pattern in this.

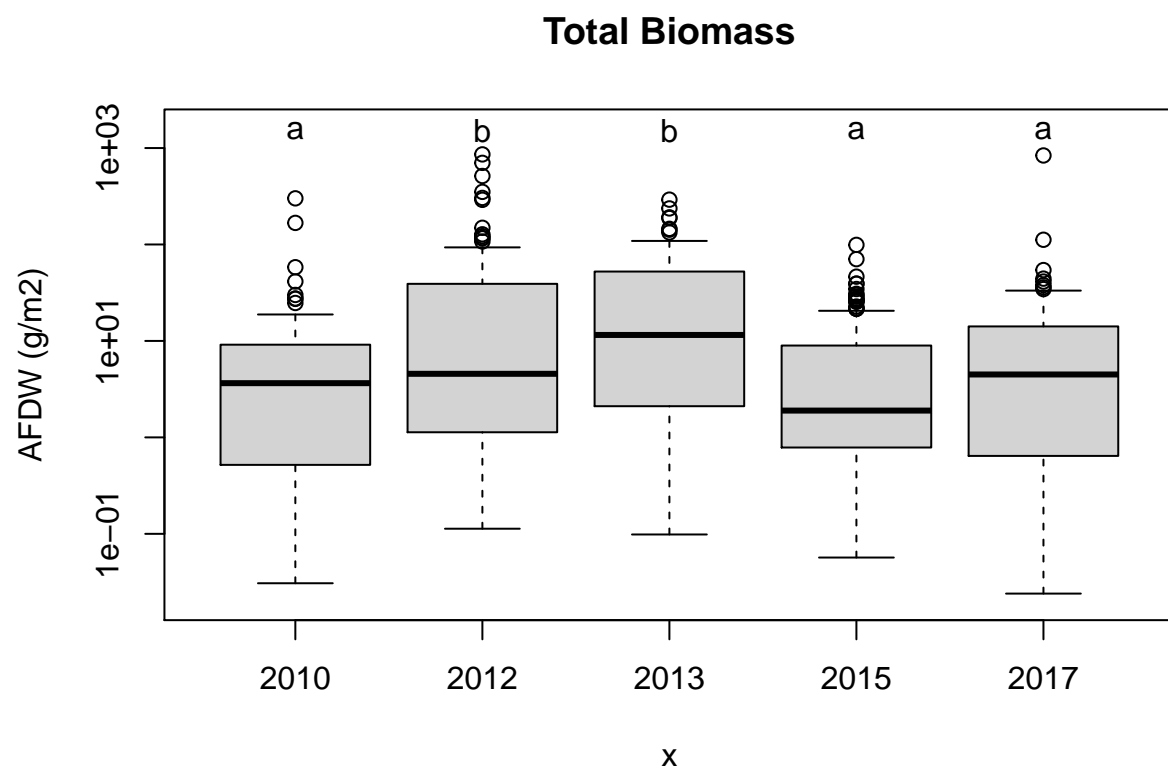
```

specnspec<-read.csv("./input/benthosdata/specnspec.csv",header=T,stringsAsFactors = FALSE)
valspecs<-which(names(dens) %in% specnspec$Taxon_name[specnspec$valid==1])
smpca$totdens<-apply(dens,1,sum)
smpca$totdens[smpca$totdens==0]<-NA
smpca$totbiom<-apply(biom,1,sum)
smpca$totbiom[smpca$totbiom==0]<-NA
smpca$nspec<-apply(dens[,valspecs],1,function(x)length(x[x>0]))
smpca$nspec[smpca$nspec==0]<-NA
ds<- smpca$totdens
bs<- smpca$totbiom
smpca$avgsz<-bs/ds
#sp_plot("Average Individual Weight",log(smpca$avgsz)+9,5)
#sp_plot("Total Biomass",log(bs)+4,10)
#sp_plot("Total Density",log(ds)+4,10)
#sp_plot("Number of species",smpca$nspec,30)
smpca$jaar<-as.factor(smpca$jaar)
plot(smpca$jaar,ds,log="y",main="Total Density",ylim=c(20,32000),ylab="Abundance (N/m2)")
text(5,30000,"a")
text(4,30000,"a")
text(3,30000,"a")
text(2,30000,"b")
text(1,30000,"a")

```

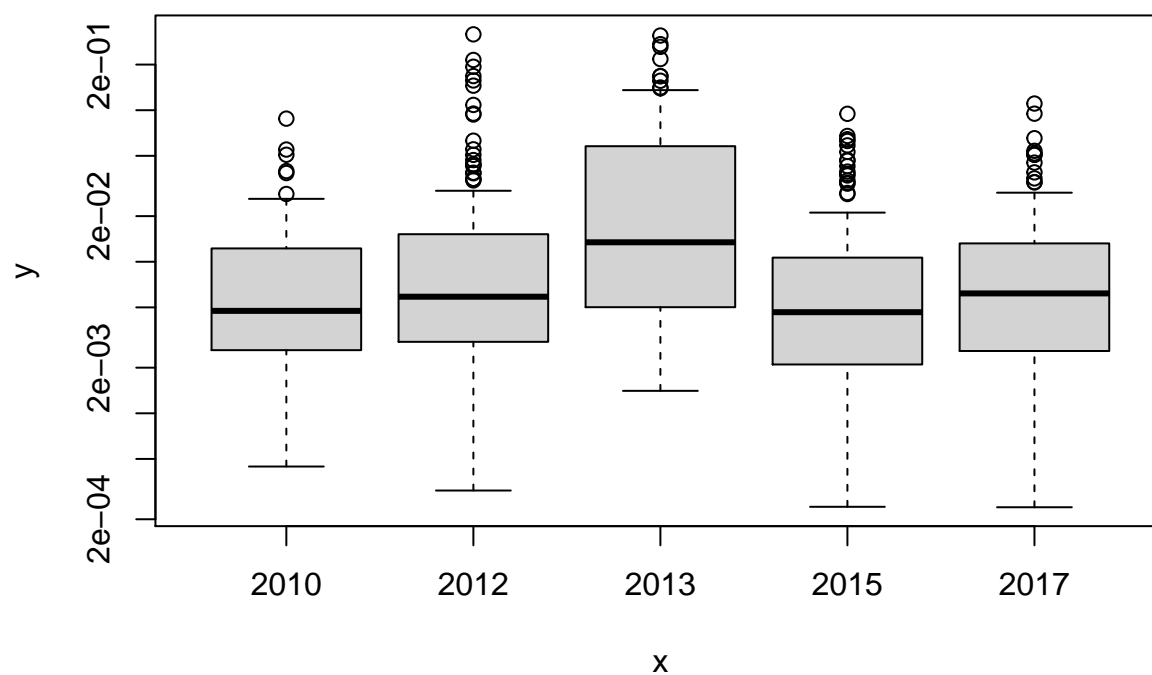


```
plot(smpca$jaar,bs,log="y",main="Total Biomass",ylim=c(0.02,1600),ylab="AFDW (g/m2)")
text(5,1500,"a")
text(4,1500,"a")
text(3,1500,"b")
text(2,1500,"b")
text(1,1500,"a")
```

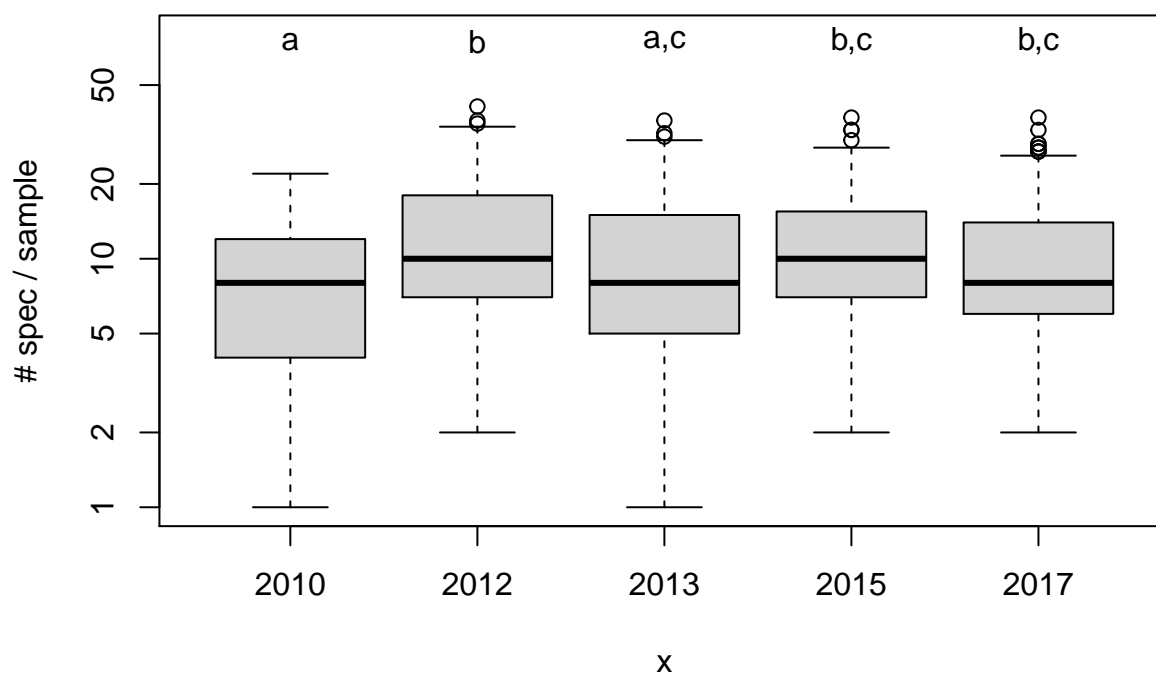
```
plot(smpca$jaarf,smpca$avgsz,log="y",main="Average Individual Weight")
```

Average Individual Weight



```
plot(smpca$jaar, smpca$nspec, log="y", main="Number of species per sample", ylab="# spec / sample",
     ylim=c(1,80))
text(1,75,"a")
text(2,75,"b")
text(3,75,"a,c")
text(4,75,"b,c")
text(5,75,"b,c")
```

Number of species per sample



```
aovtd<-aov(log(totdens)~jaarf,smpca)
posthoc<-TukeyHSD(aovtd,"jaarf")
print(posthoc)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = log(totdens) ~ jaarf, data = smpca)
##
## $jaarf
##          diff          lwr          upr      p adj
## 2012-2010  0.67624724  0.1222310  1.2302635164 0.0079294
## 2013-2010  0.14014643 -0.4355805  0.7158733482 0.9634754
## 2015-2010  0.22085950 -0.3331568  0.7748757777 0.8110380
## 2017-2010  0.12840792 -0.4256084  0.6824242020 0.9694084
## 2013-2012 -0.53610081 -1.0171530 -0.0550486645 0.0202174
## 2015-2012 -0.45538774 -0.9102325 -0.0005430217 0.0495582
## 2017-2012 -0.54783931 -1.0026840 -0.0929945974 0.0091972
## 2015-2013  0.08071307 -0.4003391  0.5617652186 0.9908260
## 2017-2013 -0.01173850 -0.4927906  0.4693136428 0.9999955
## 2017-2015 -0.09245158 -0.5472963  0.3623931413 0.9811199
```

```
aovtb<-aov(log(totbiom)~jaarf,smpca)
posthoc<-TukeyHSD(aovtb,"jaarf")
print(posthoc)
```

```
## Tukey multiple comparisons of means
```

```
##      95% family-wise confidence level
##
## Fit: aov(formula = log(totbiom) ~ jaarf, data = smpca)
##
## $jaarf
##              diff              lwr              upr              p adj
## 2012-2010  0.8917829  0.05214959  1.73141621  0.0310071
## 2013-2010  1.2987275  0.42480941  2.17264553  0.0005247
## 2015-2010  0.1169426 -0.72269072  0.95657591  0.9955084
## 2017-2010  0.2936659 -0.55202979  1.13936166  0.8767804
## 2013-2012  0.4069446 -0.32019921  1.13408835  0.5420113
## 2015-2012 -0.7748403 -1.46039803 -0.08928258  0.0176929
## 2017-2012 -0.5981170 -1.29108635  0.09485241  0.1273217
## 2015-2013 -1.1817849 -1.90892865 -0.45464109  0.0001033
## 2017-2013 -1.0050615 -1.73919724 -0.27092583  0.0018527
## 2017-2015  0.1767233 -0.51624604  0.86969272  0.9568484
```

```
aovas<-aov(log(avgsized)~jaarf,smpca)
posthoc<-TukeyHSD(aovas,"jaarf")
print(posthoc)
```

```
##      Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = log(avgsized) ~ jaarf, data = smpca)
##
## $jaarf
##              diff              lwr              upr              p adj
## 2012-2010  0.23084176 -0.3385780  0.8002615  0.8013681
## 2013-2010  1.14778890  0.5551180  1.7404598  0.0000017
## 2015-2010 -0.08861081 -0.6580306  0.4808090  0.9931166
## 2017-2010  0.10613136 -0.4673998  0.6796625  0.9867066
## 2013-2012  0.91694714  0.4238152  1.4100791  0.0000050
## 2015-2012 -0.31945257 -0.7843819  0.1454767  0.3288839
## 2017-2012 -0.12471040 -0.5946661  0.3452453  0.9503250
## 2015-2013 -1.23639970 -1.7295317 -0.7432677  0.0000000
## 2017-2013 -1.04165754 -1.5395313 -0.5437838  0.0000002
## 2017-2015  0.19474217 -0.2752135  0.6646979  0.7881993
```

```
aovns<-aov(log(nspec)~jaarf,smpca)
posthoc<-TukeyHSD(aovns,"jaarf")
print(posthoc)
```

```
##      Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = log(nspec) ~ jaarf, data = smpca)
##
## $jaarf
##              diff              lwr              upr              p adj
## 2012-2010  0.43873540  0.14558706  0.73188374  0.0004656
## 2013-2010  0.16376425 -0.14087190  0.46840041  0.5814546
## 2015-2010  0.39099220  0.09784386  0.68414053  0.0026557
## 2017-2010  0.30713878  0.01399044  0.60028712  0.0347345
## 2013-2012 -0.27497115 -0.52951175 -0.02043055  0.0267987
```

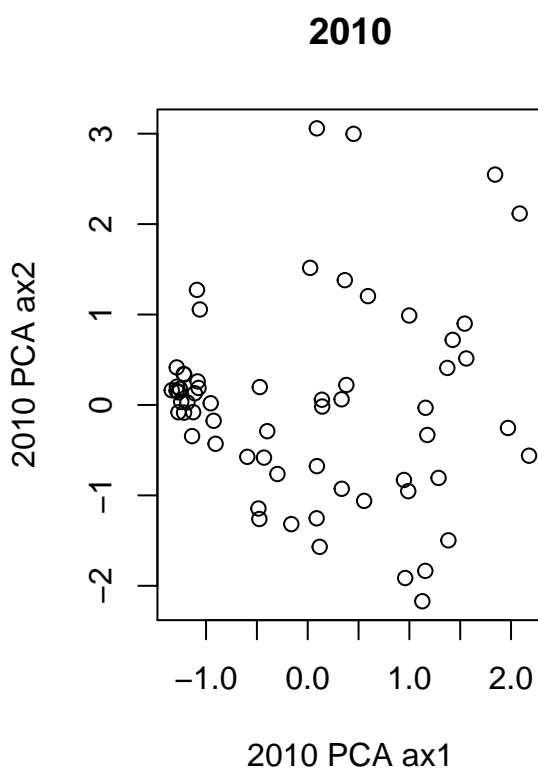
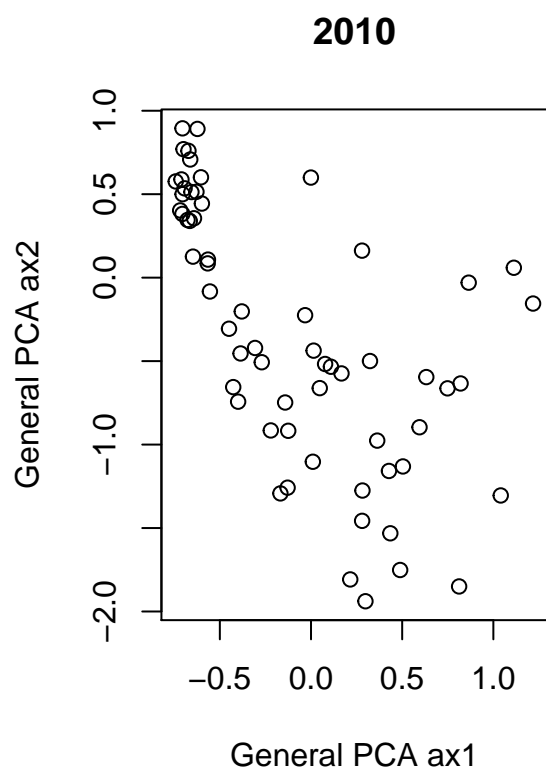
```
## 2015-2012 -0.04774320 -0.28841659 0.19293018 0.9827548
## 2017-2012 -0.13159662 -0.37227000 0.10907676 0.5650069
## 2015-2013 0.22722794 -0.02731266 0.48176854 0.1055266
## 2017-2013 0.14337453 -0.11116607 0.39791513 0.5356045
## 2017-2015 -0.08385342 -0.32452680 0.15681997 0.8754732
```

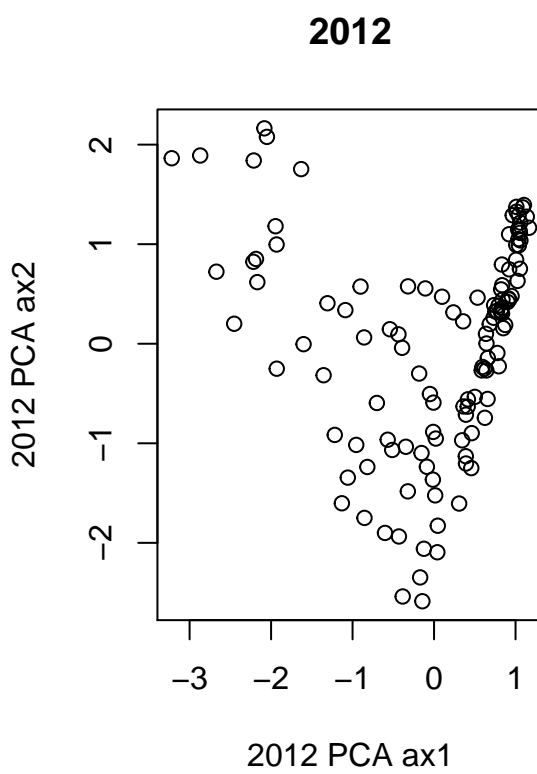
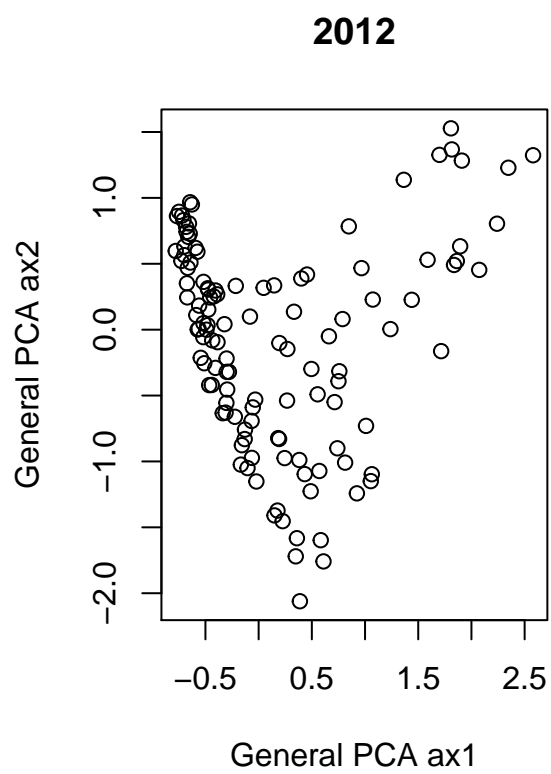
Multivariate analysis of the benthos data

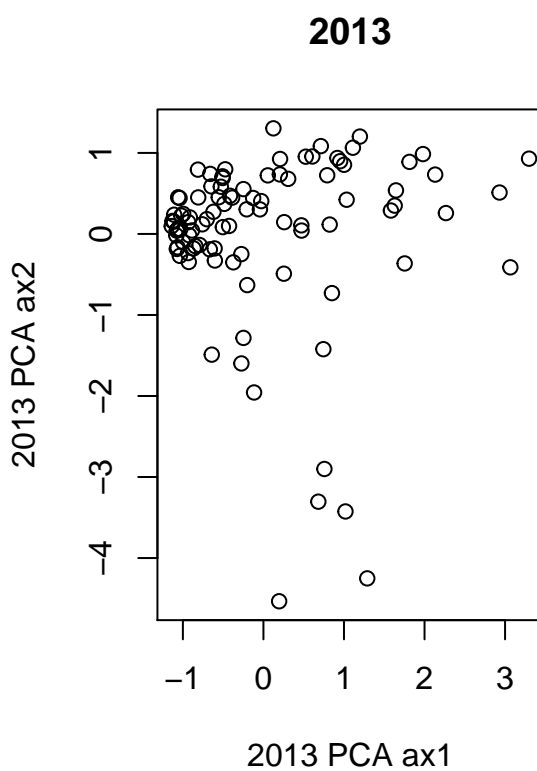
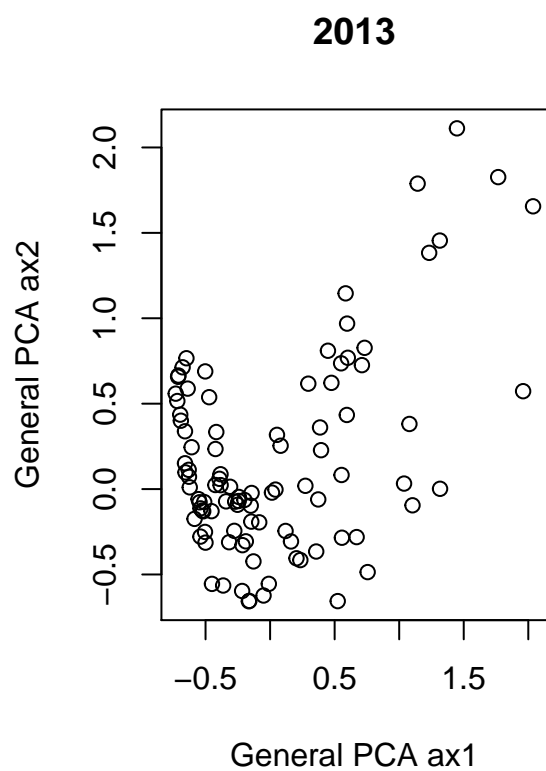
We start the analysis with a pca on the log-transformed density data. Only the first axis of this analysis contains real information. The second axis is quadratically related to the first, as is often the case when only one dominant environmental gradient is present.

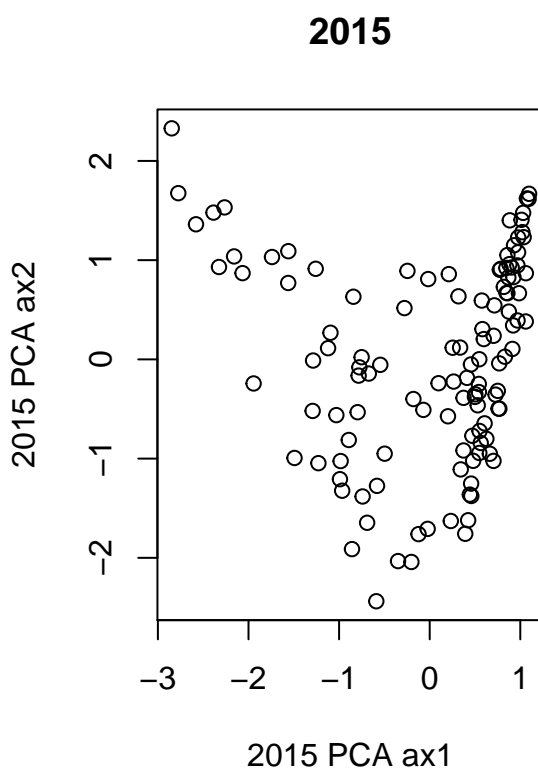
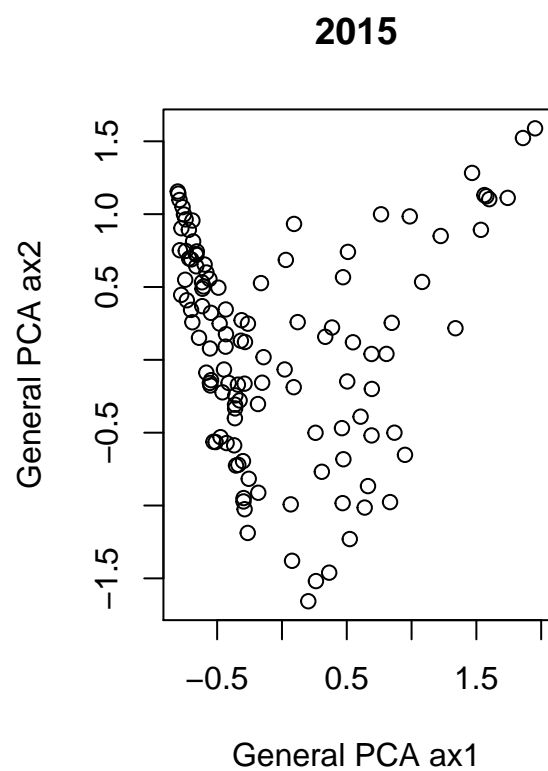
In order to check on the general applicability of this pca to represent major variations in the different years, we have also performed a pca on each of the years separately. In general, these pca analyses correlated reasonably to very well with the result of the overall pca for the year. Therefore these separate analyses are not used any further, as they all subtly differ and are therefore difficult to intercompare.

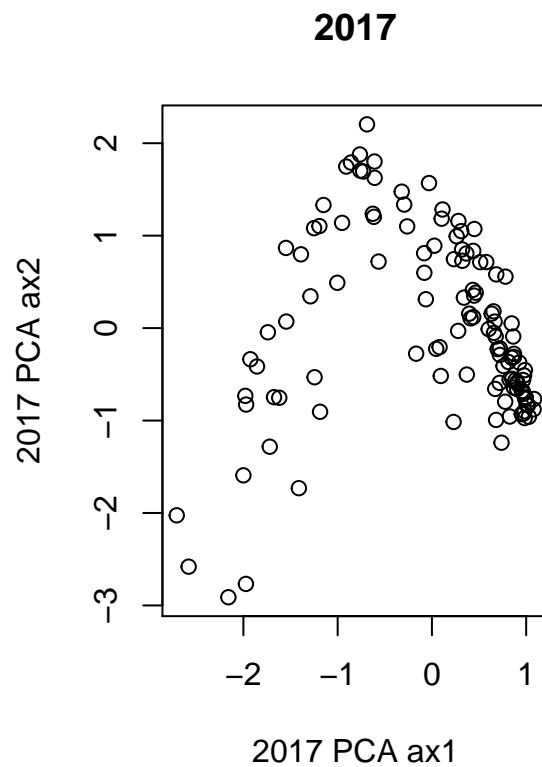
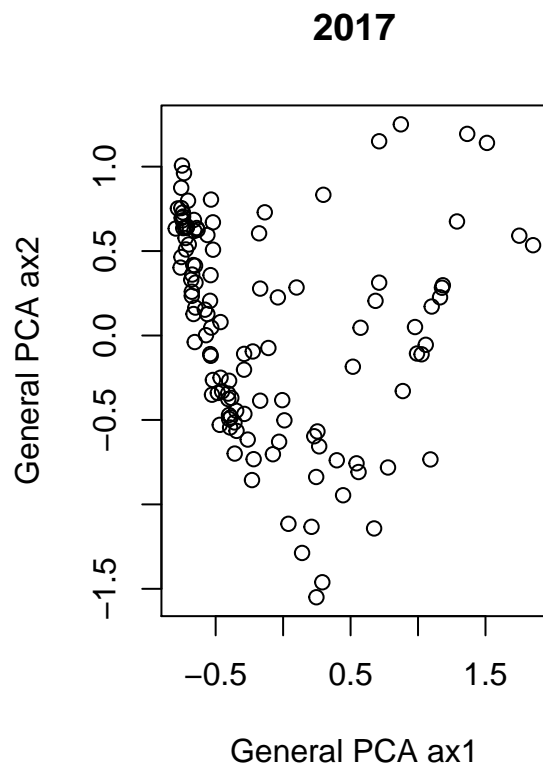
```
pngbasedir<-"./output/pngs/dens/"
##### pca of all benthos for all years #####
ldens<-log(dens+1)
#lbiom<-log(biom*1000+1)
row.names(ldens)<-stats$StationNr
pca<-rda(ldens)
smpca$ax1<-scores(pca)$sites[,1]
smpca$ax2<-scores(pca)$sites[,2]
##### pca per year #####
par(mfrow=c(1,2))
for (jr in c(2010,2012,2013,2015,2017)){
  xm<-ldens[stats$jaar==jr,]
  jr pca<-rda(xm)
  smpca$jrax1[stats$jaar==jr]<-scores(jrpca)$sites[,1]
  smpca$jrax2[stats$jaar==jr]<-scores(jrpca)$sites[,2]
  plot(smpca$ax1[stats$jaar==jr],smpca$ax2[stats$jaar==jr],main=paste(jr),
        xlab="General PCA ax1",ylab="General PCA ax2")
  plot(smpca$jrax1[stats$jaar==jr],smpca$jrax2[stats$jaar==jr],main=paste(jr),
        xlab=paste(jr,"PCA ax1"),ylab=paste(jr,"PCA ax2"))
}
```









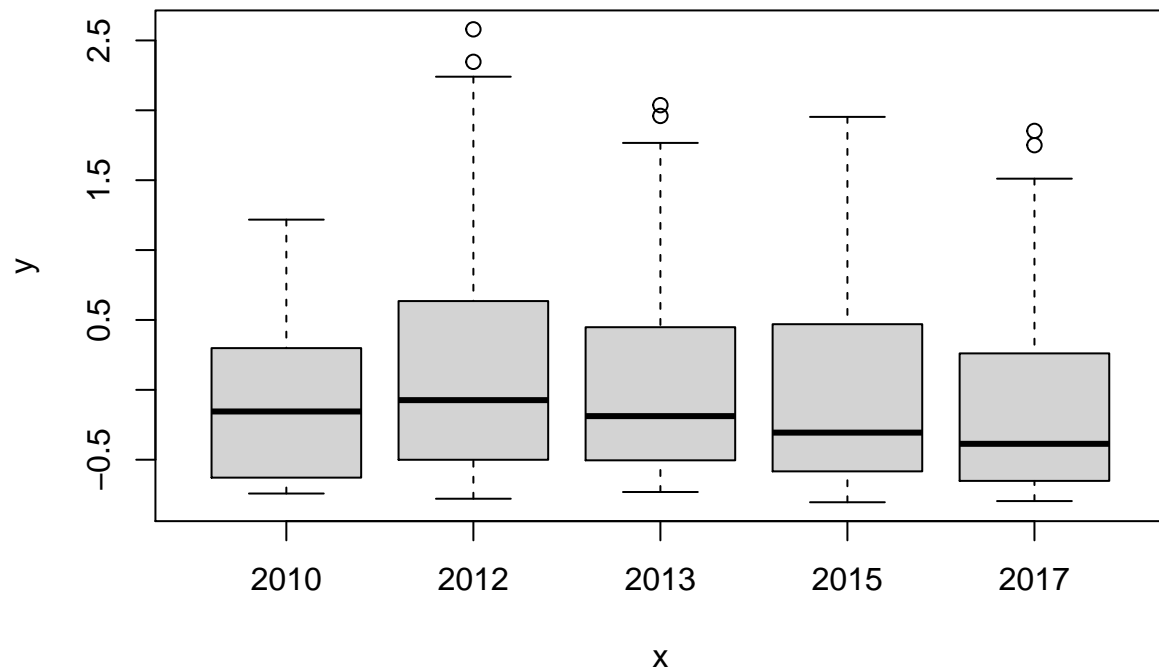


```
par(mfrow=c(1,1))

# transform pca1 values to make them more linearly dependent on depth
smpca$a1t<- log(1+smpca$ax1)
smpca$jaarf<-as.factor(smpca$jaar)

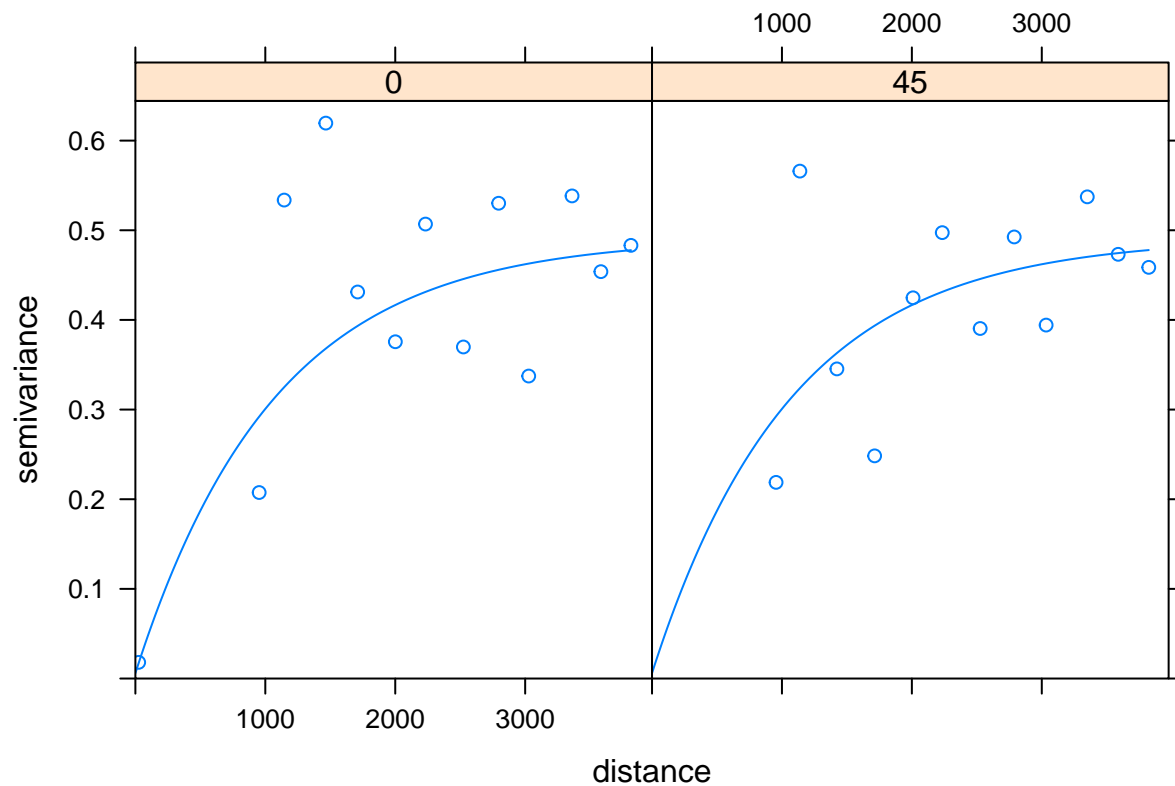
#make boxplot of axis1 scores per year
plot(smpca$jaarf,smpca$a1t,main="Scores first PCA axis")
```

Scores first PCA axis



```
##### spatially interpolate axis1 observations #####
tc<-terrain.colors(256)
bluered<-colorRampPalette(c("darkblue","white","darkred"))(256)

vzd <- variogram(alt~1, smpca[smpca$jaar==2010,],alpha=c(0,45))
mzd <- fit.variogram(vzd, vgm(.1, "Exp", 1000,0.1,anis=c(45,1)))
plot(vzd,mzd)
```



```
for (jr in c(2010,2012,2013,2015,2017)){
  welk<-which(smpca$jaar==jr)
  mgs<-smpca[welk,]
  gzd <- gstat(NULL, "GS", alt~1, mgs, model=mzd)
  raltst <- interpolate(r, gzd)
  raltst <- mask(raltst,mgrid)
  names(raltst)<-paste("altst_",jr,sep="")
  barnam<-paste("bath_",jr,sep="")
  bar<-subset(b,barnam)
  bar[bar< -1]<- NA
  raltst<-mask(raltst,bar)

  png(filename=paste(pngbasedir,"obsax1_",jr,".png",sep=""),
       width=1000,height=1000,pointsize=24,type="windows")
  plot(raltst,zlim=c(-1.75,1.75),
       main=paste("Observed axis1 scores for year",jr),
       col=bluered)
  cls<-bluered[(mgs$alt+1.75)/3.5*255]
  points(coordinates(mgs),pch=19,col=cls)
  dev.off()
}
```

```
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
```

```
## [using ordinary kriging]
## [using ordinary kriging]
# make compound png out of year plots
compplot("obsax1")
##### END interpolation observations #####
```

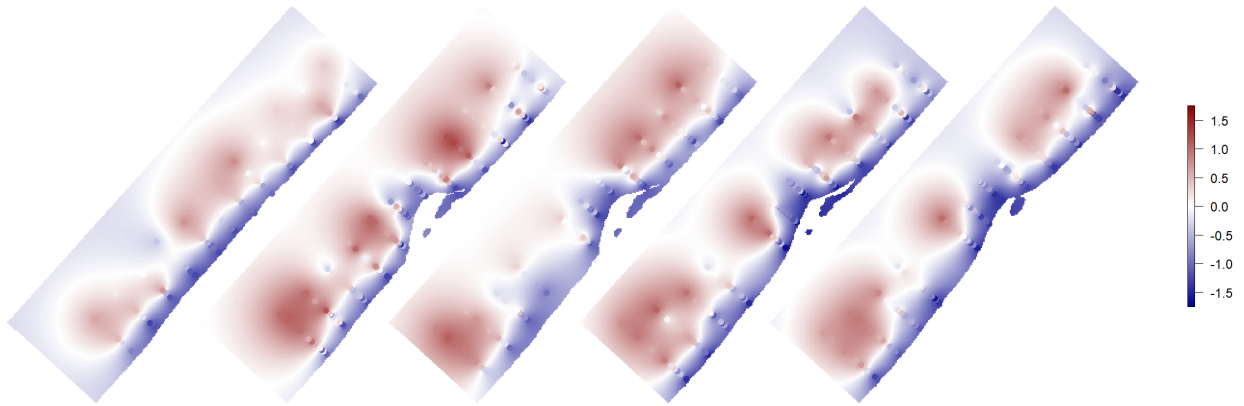


Figure 1: observed axis 1 scores

Linear model for PCA scores

In this step of the analysis, a linear model is constructed to explain the variation between samples in scores on the first pca axis. This model is based on the relations in the ‘normal’ linear coast, as observed in 2010. A fairly good statistical model for the scores can be constructed based on grain size and the interaction between depth and total bottom shear stress due to currents (mainly tide) and waves.

We use this model to predict the scores in the subsequent years, and note the differences between these predictions and the observations. Attention is focused on the spatial pattern of these differences, which are spatially interpolated.

For comparison with these plots, we also produce plots of erosion/deposition, based on the subsequent bathymetries.

```
##### simple linear regression model based on 2010 #####
# transform pca1 values to make them more linearly dependent on depth
smpca$a1t<- log(1+smpca$ax1)
# perform simple correlation analysis with some env variables
lmtt<-lm(a1t~sd50um,smpca)
summary(lmtt)
```

```
##
## Call:
## lm(formula = a1t ~ sd50um, data = smpca)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.42235 -0.57501 -0.00323  0.57442  1.60094
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.2062638  0.0886847   2.326   0.0204 *
## sd50um       -0.0015521  0.0002848  -5.449 7.85e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6952 on 517 degrees of freedom
## Multiple R-squared:  0.05431,    Adjusted R-squared:  0.05248
## F-statistic: 29.69 on 1 and 517 DF,  p-value: 7.851e-08
```

```
lmtt<-lm(a1t~depth,smpca)
summary(lmtt)
```

```
##
## Call:
## lm(formula = a1t ~ depth, data = smpca)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11717 -0.32787 -0.00346  0.33780  1.31685
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.329445   0.046349  -28.68  <2e-16 ***
## depth        0.157178   0.006029   26.07  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4699 on 517 degrees of freedom
## Multiple R-squared:  0.5679, Adjusted R-squared:  0.5671
## F-statistic: 679.6 on 1 and 517 DF,  p-value: < 2.2e-16
```

```
lmtt<-lm(a1t~log(tt),smpca)
summary(lmtt)
```

```
##
## Call:
## lm(formula = a1t ~ log(tt), data = smpca)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56320 -0.43532 -0.03041  0.45701  1.33801
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.89471    0.04926  -18.16  <2e-16 ***
## log(tt)      -0.81573    0.05277  -15.46  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5913 on 517 degrees of freedom
## Multiple R-squared:  0.3161, Adjusted R-squared:  0.3147
## F-statistic: 238.9 on 1 and 517 DF,  p-value: < 2.2e-16
```

```
# perform linear regression
```

```
lmdg<-lm(a1t~sd50um+depth~log(tt),smpca[smpca$jaar==2010,])
summary(lmdg)
```

```
##
```

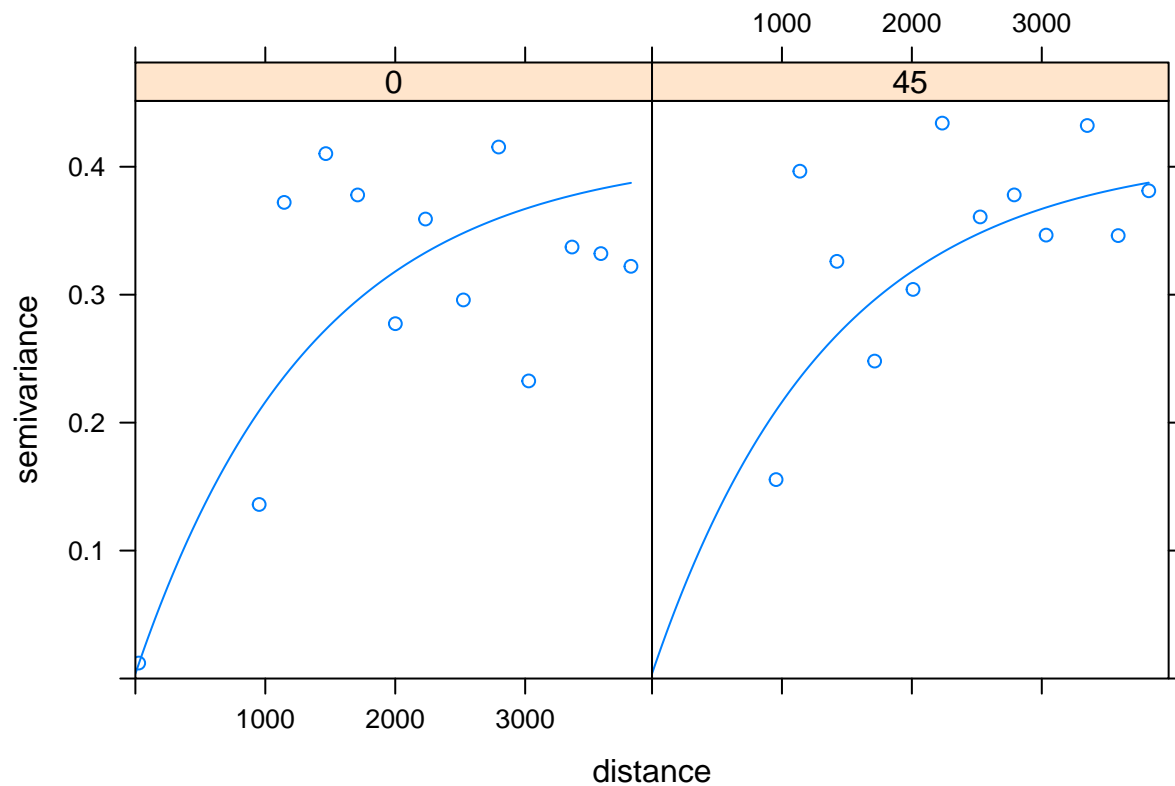


```
## Call:
## lm(formula = a1t ~ sd50um + depth:log(tt), data = smpca[smpca$jaar ==
##      2010, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7858 -0.2229 -0.0136  0.2434  0.7173
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.660954   0.175966  -3.756 0.000397 ***
## sd50um        -0.002498   0.000598  -4.177 9.9e-05 ***
## depth:log(tt) -0.159992   0.011635 -13.751 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3139 on 59 degrees of freedom
## Multiple R-squared:  0.7843, Adjusted R-squared:  0.777
## F-statistic: 107.3 on 2 and 59 DF,  p-value: < 2.2e-16

smpca$pred<-predict(lmdg,newdata=smpca)
smpca$dev<-smpca$a1t-smpca$pred
smpca$devst<-smpca$dev

##### spatially interpolate predictions #####
tc<-terrain.colors(256)
bluered<-colorRampPalette(c("darkblue","white","darkred"))(256)

vzd <- variogram(pred~1, smpca[smpca$jaar==2010,],alpha=c(0,45))
mzd <- fit.variogram(vzd, vgm(.1, "Exp", 1000,0.1,anis=c(45,1)))
plot(vzd,mzd)
```



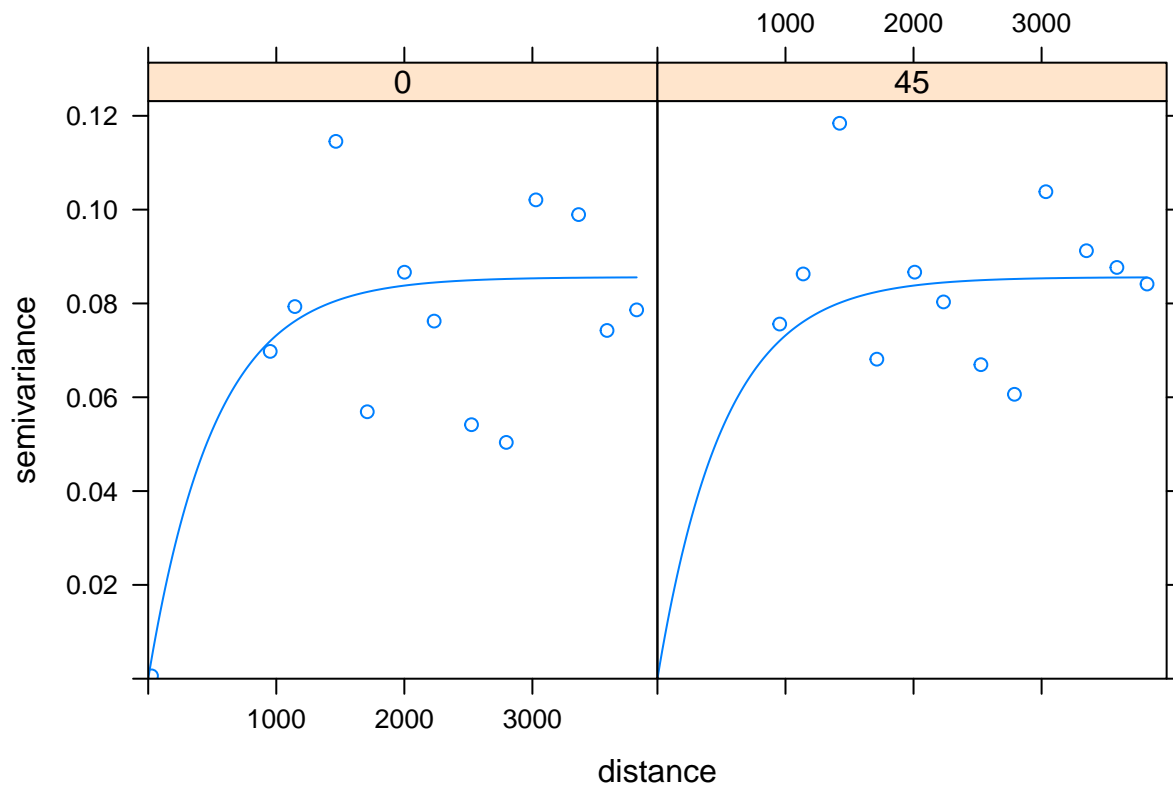
```
for (jr in c(2010,2012,2013,2015,2017)){
  welk<-which(smpca$jaar==jr)
  mgs<-smpca[welk,]
  gzd <- gstat(NULL, "GS", pred~1, mgs, model=mzd)
  rpredst <- interpolate(r, gzd)
  rpredst <- mask(rpredst,mgrid)
  names(rpredst)<-paste("predst_",jr,sep="")
  barnam<-paste("bath_",jr,sep="")
  bar<-subset(b,barnam)
  bar[bar< -1]<- NA
  rpredst<-mask(rpredst,bar)
  png(filename=paste(pngbasedir,"predax1_",jr,".png",sep=""),
       width=1000,height=1000,pointsize=24,type="windows")
  plot(rpredst,zlim=c(-1.75,1.75),
       main=paste("Prediction based on 2010 for year",jr),
       col=bluered)
  cls<-bluered[(mgs$predst+1.75)/3.5*255]
  points(coordinates(mgs),pch=19,col=cls)
  dev.off()
}
```

```
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
```

```
## [using ordinary kriging]
compplot("predax1")
##### END interpolation predictions #####

##### spatially interpolate deviations #####
tc<-terrain.colors(256)
bluered<-colorRampPalette(c("darkblue","white","darkred"))(256)

vzd <- variogram(devst~1, smpca[smpca$jaar==2010,],alpha=c(0,45))
mzd <- fit.variogram(vzd, vgm(.1, "Exp", 1000,0.1,anis=c(45,1)))
plot(vzd,mzd)
```



```
for (jr in c(2010,2012,2013,2015,2017)){
  welk<-which(smpca$jaar==jr)
  mgs<-smpca[welk,]
  gzd <- gstat(NULL, "GS", devst~1, mgs, model=mzd)
  rdevst <- interpolate(r, gzd)
  rdevst <- mask(rdevst,mgrid)
  names(rdevst)<-paste("devst_",jr,sep="")
  #b<-brick(rdevst,b)
  barnam<-paste("bath_",jr,sep="")
  bar<-subset(b,barnam)
  bar[bar<-1]<- NA
  rdevst<-mask(rdevst,bar)
  png(filename=paste(pngbasedir,"devax1_",jr,".png",sep=""),
      width=1000,height=1000,points=24,type="windows")
}
```

```

plot(rdevst,zlim=c(-1.5,1.5),main=paste(jr),col=bluered)
cls<-bluered[(mgs$devst+1.5)/3*255]
points(coordinates(mgs),pch=19,col=cls)
dev.off()
}

## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
compplot("devax1")

##### END interpolation deviations #####
#
##### simple linear regression model based on all years #####
# perform linear regression
lmdga<-lm(a1t~sd50um+depth:log(tt),smpca)
summary(lmdga)

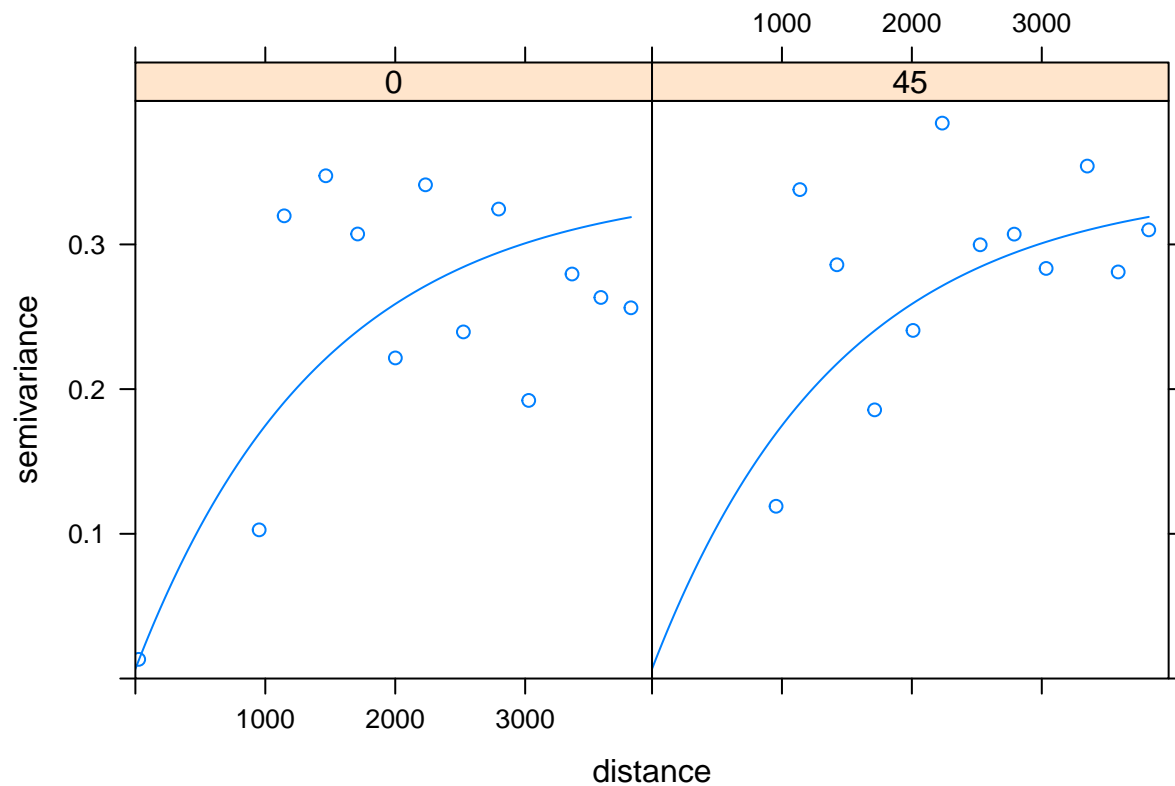
##
## Call:
## lm(formula = a1t ~ sd50um + depth:log(tt), data = smpca)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09682 -0.28547  0.01552  0.24340  1.04527
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.8060266  0.0560044 -14.392  <2e-16 ***
## sd50um        -0.0013830  0.0001545  -8.951  <2e-16 ***
## depth:log(tt) -0.1514081  0.0042949 -35.253  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3769 on 516 degrees of freedom
## Multiple R-squared:  0.7225, Adjusted R-squared:  0.7215
## F-statistic: 671.9 on 2 and 516 DF,  p-value: < 2.2e-16

smpca$preda<-predict(lmdga,newdata=smpca)
smpca$deva<-smpca$a1t-smpca$preda

##### spatially interpolate predictions #####
tc<-terrain.colors(256)
bluered<-colorRampPalette(c("darkblue","white","darkred"))(256)

vzd <- variogram(preda~1, smpca[smpca$jaar==2010,],alpha=c(0,45))
mzd <- fit.variogram(vzd, vgm(.1, "Exp", 1000,0.1,anis=c(45,1)))
plot(vzd,mzd)

```



```
for (jr in c(2010,2012,2013,2015,2017)){
  welk<-which(smpca$jaar==jr)
  mgs<-smpca[welk,]
  gzd <- gstat(NULL, "GS", pred=1, mgs, model=mzd)
  rpreda <- interpolate(r, gzd)
  rpreda <- mask(rpreda,mgrid)
  names(rpreda)<-paste("preda_",jr,sep="")
  b<-brick(rpreda,b)
  barnam<-paste("bath_",jr,sep="")
  bar<-subset(b,barnam)
  bar[bar< -1]<- NA
  rpreda<-mask(rpreda,bar)

  png(filename=paste(pngbasedir,"predfromallax1_",jr,".png",sep=""),
       width=1000,height=1000,pointsize=24,type="windows")
  plot(rpreda,zlim=c(-1.75,1.75),
       main=paste("Prediction based on all years for year",jr),
       col=bluered)
  cls<-bluered[(mgs$preda+1.75)/3.5*255]
  points(coordinates(mgs),pch=19,col=cls)
  dev.off()
}
```

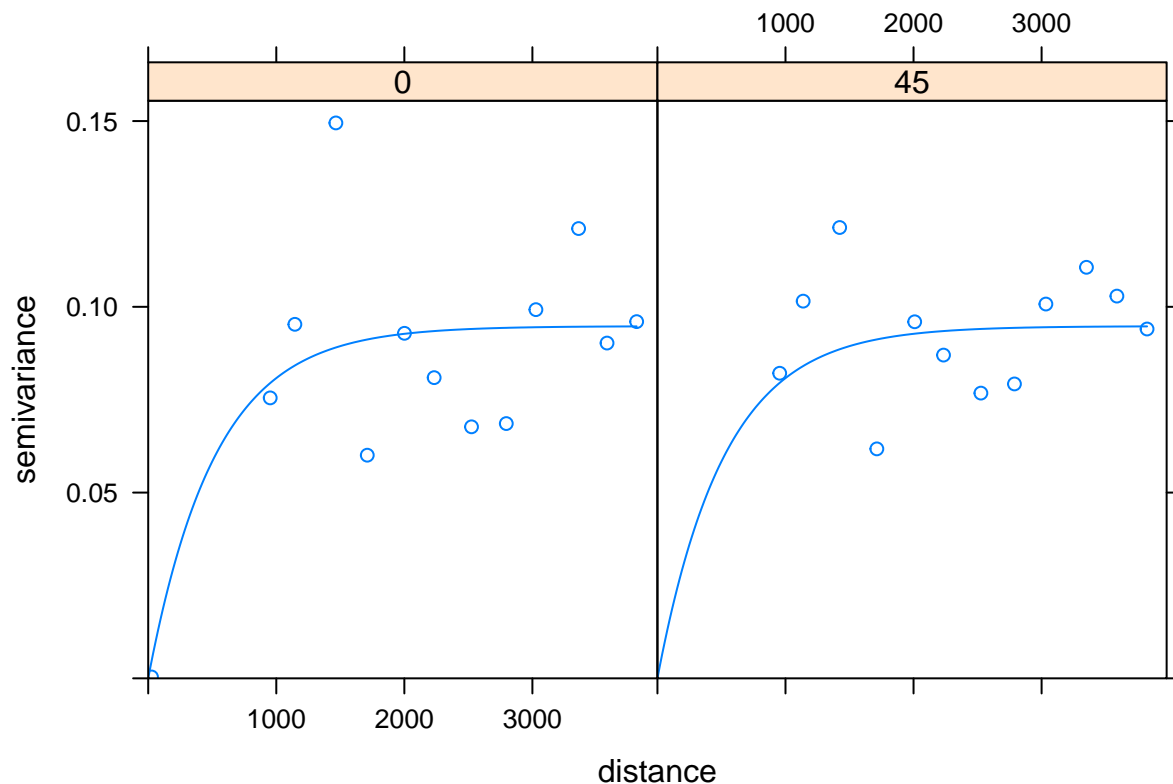
```
## [using ordinary kriging]
```

```
## [using ordinary kriging]
```

```
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
compplot("predfromallax1")
##### END interpolation predictions #####

##### spatially interpolate deviations #####
tc<-terrain.colors(256)
bluered<-colorRampPalette(c("darkblue","white","darkred"))(256)

vzd <- variogram(deva~1, smpca[smpca$jaar==2010,],alpha=c(0,45))
mzd <- fit.variogram(vzd, vgm(.1, "Exp", 1000,0.1,anis=c(45,1)))
plot(vzd,mzd)
```



```
for (jr in c(2010,2012,2013,2015,2017)){
  welk<-which(smpca$jaar==jr)
  mgs<-smpca[welk,]
  gzd <- gstat(NULL, "GS", deva~1, mgs, model=mzd)
  rdeva <- interpolate(r, gzd)
  rdeva <- mask(rdeva,mgrid)
  names(rdeva)<-paste("deva_",jr,sep="")
  #b<-brick(rdeva,b)
  barnam<-paste("bath_",jr,sep="")
  bar<-subset(b,barnam)
  bar[bar< -1]<- NA
}
```

```

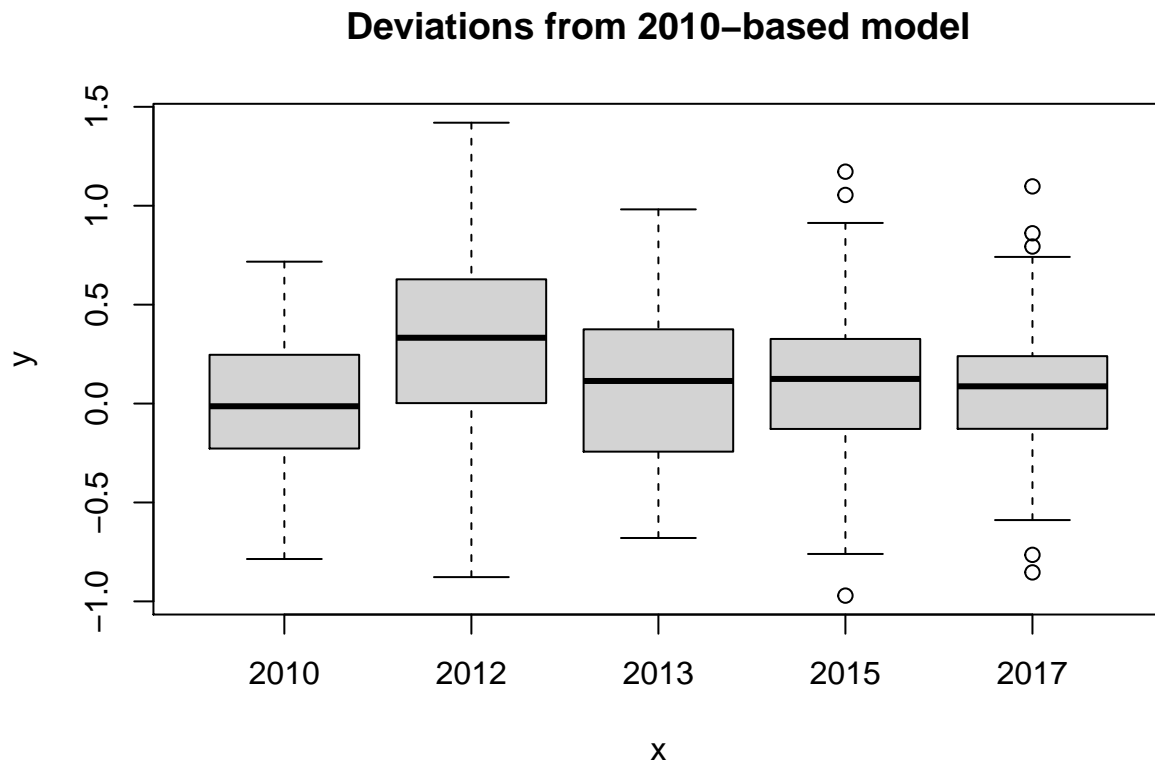
rdeva<-mask(rdeva,bar)
#repeat plotting on a png file
png(filename=paste(pngbasedir,"devfromallax1_",jr,".png",sep=""),
     width=1000,height=1000,pointsize=24,type="windows")
plot(rdeva,zlim=c(-1.5,1.5),
     main=paste("deviation from all-year model for year",jr),
     col=bluered)
cls<-bluered[(mgs$deva+1.5)/3*255]
points(coordinates(mgs),pch=19,col=cls)
dev.off()
}

## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]
## [using ordinary kriging]

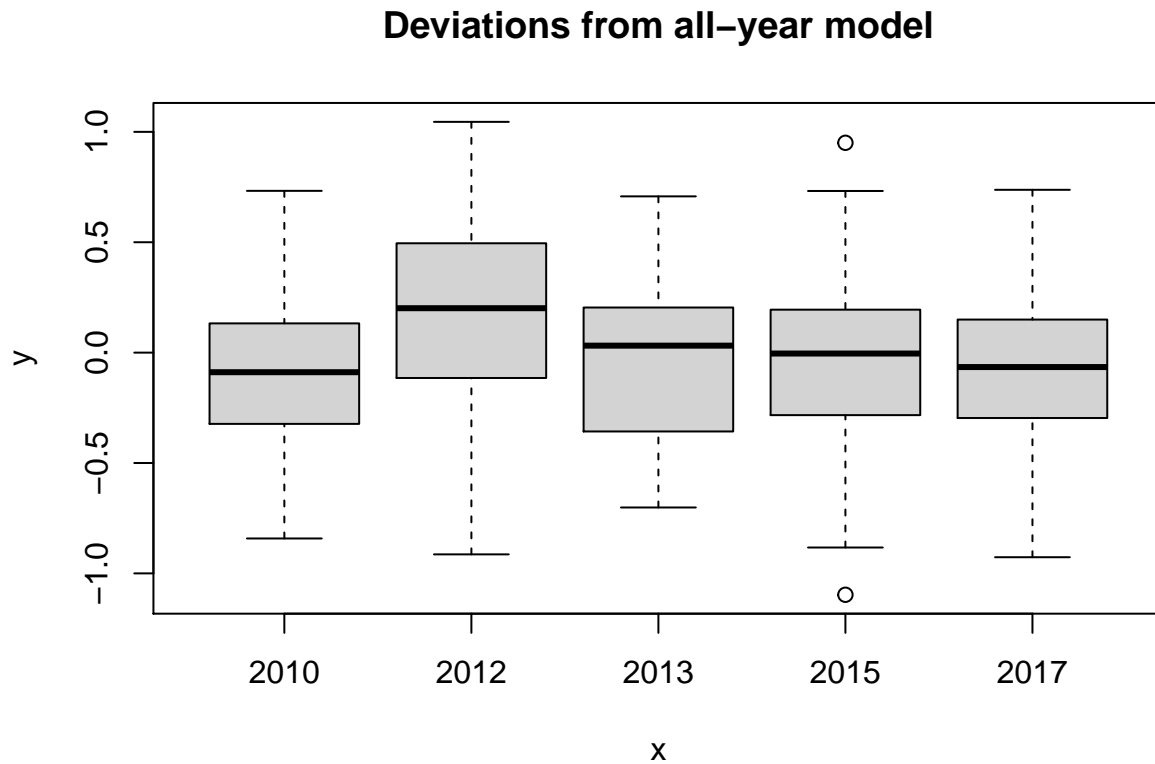
compplot("devfromallax1")
##### END interpolation deviations #####

##### year differences #####
smpca$jaarf<-as.factor(smpca$jaar)
plot(smpca$jaarf,smpca$devst,main="Deviations from 2010-based model")

```



```
plot(smpca$jaarf,smpca$deva,main="Deviations from all-year model")
```



```
lmjdt<-lm(a1t~sd50um+depth:log(tt)+log(mnf)+jaarf,smpca)
summary(lmjdt)
```

```
##
## Call:
## lm(formula = a1t ~ sd50um + depth:log(tt) + log(mnf) + jaarf,
##     data = smpca)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05592 -0.24530  0.01803  0.23760  0.98697
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4889538  0.1171100  -4.175 3.50e-05 ***
## sd50um        -0.0014557  0.0001509  -9.648 < 2e-16 ***
## log(mnf)       0.2677061  0.0635113   4.215 2.95e-05 ***
## jaarf2012     0.2386061  0.0565691   4.218 2.92e-05 ***
## jaarf2013     0.0168844  0.0592980   0.285  0.776
## jaarf2015     0.0477882  0.0567933   0.841  0.400
## jaarf2017    -0.0079184  0.0568101  -0.139  0.889
## depth:log(tt) -0.1513770  0.0041237 -36.709 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##
## Residual standard error: 0.3594 on 511 degrees of freedom
## Multiple R-squared:  0.7503, Adjusted R-squared:  0.7468
## F-statistic: 219.3 on 7 and 511 DF,  p-value: < 2.2e-16

# save smpca
save(smpca,file="./output/binary_files/smpca.Rdata")
```

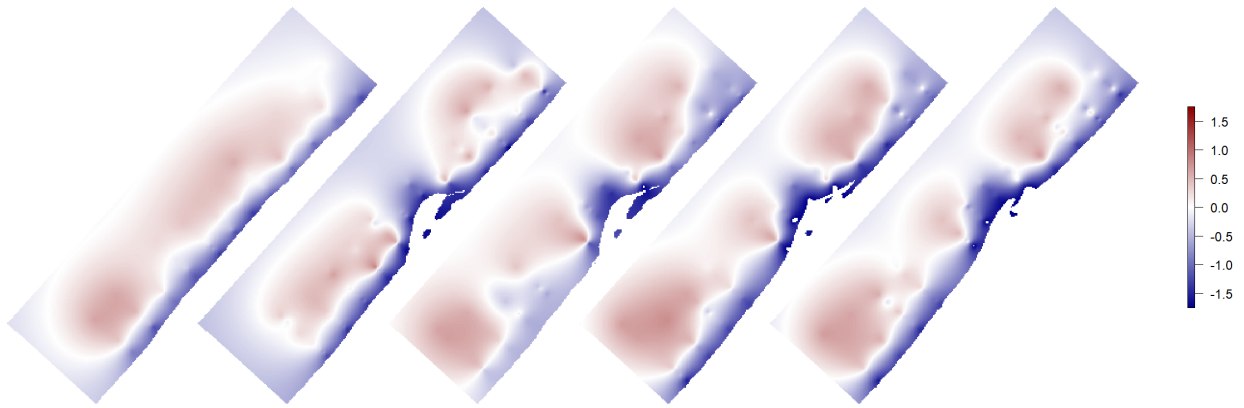


Figure 2: predicted axis 1 scores

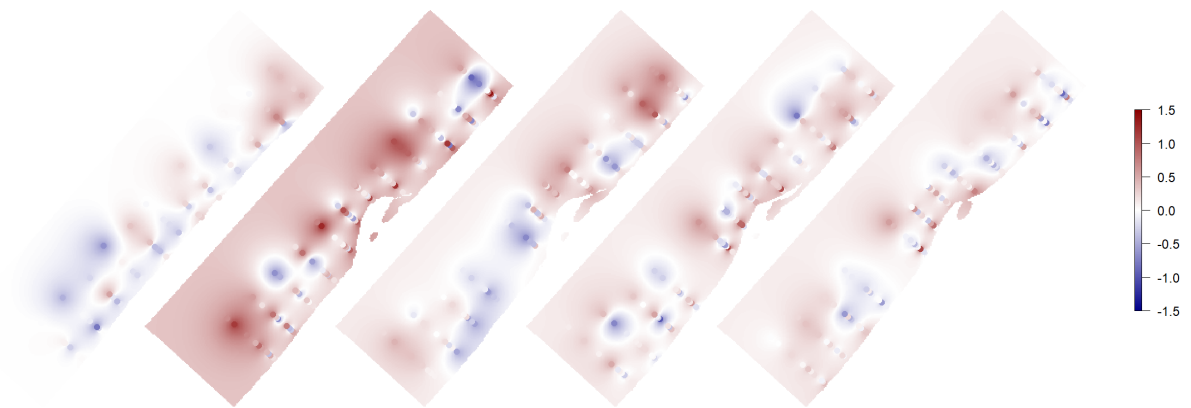


Figure 3: deviations axis 1 scores

Relations between depth, bottom shear stress, grain size and fauna

Here we make plots for the different years of the scores on the first pca axis (log-transformed), while indicating in the graph either the bottom shear stress or the grain size. It shows how the relations, that are fixed along a linear coast (especially for bottom shear stress) are modified due to the construction of the sand motor. Different combinations appear, that have consequences for the fauna.

```
for (jr in c(2010,2012,2013,2015,2017)){
  symb<-smpca$sd50um[smpca$jaar==jr]
  cls<-terrain.colors(256)[(symb-90)/630*256]
  plot(smpca$dep_cm[smpca$jaar==jr],smpca$a1t[smpca$jaar==jr],
       cex=symb/300,col=cls,pch=19,
       main=paste(jr,"symbols=med gr size"),
       xlab="Depth (cm)",ylab="Transformed Axis 1",
       xlim=c(-1200,0),ylim=c(-2.1,1.5))
}
```

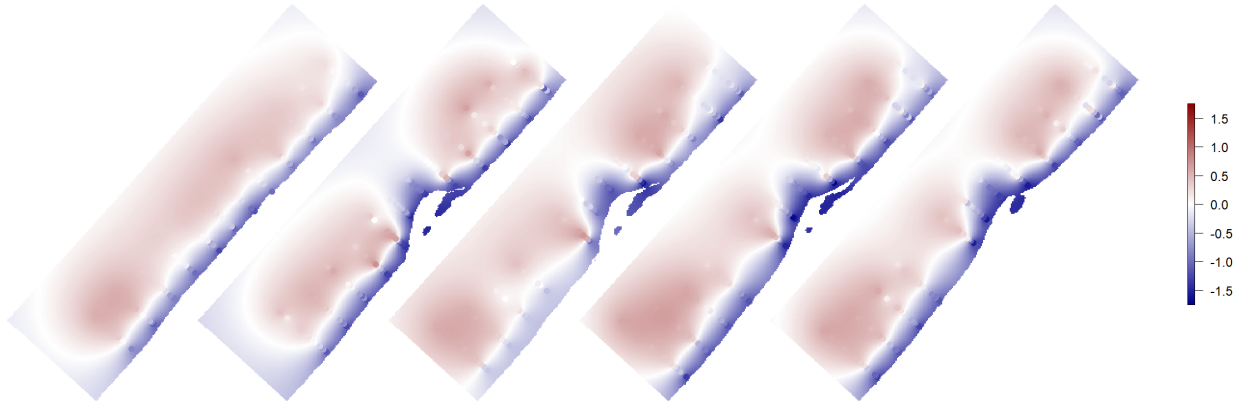


Figure 4: prediction based on all axis 1 scores

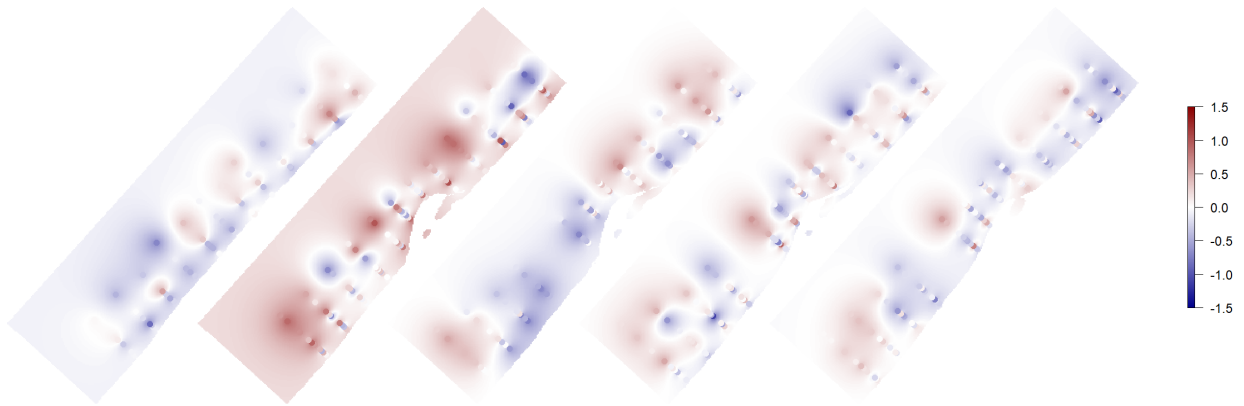
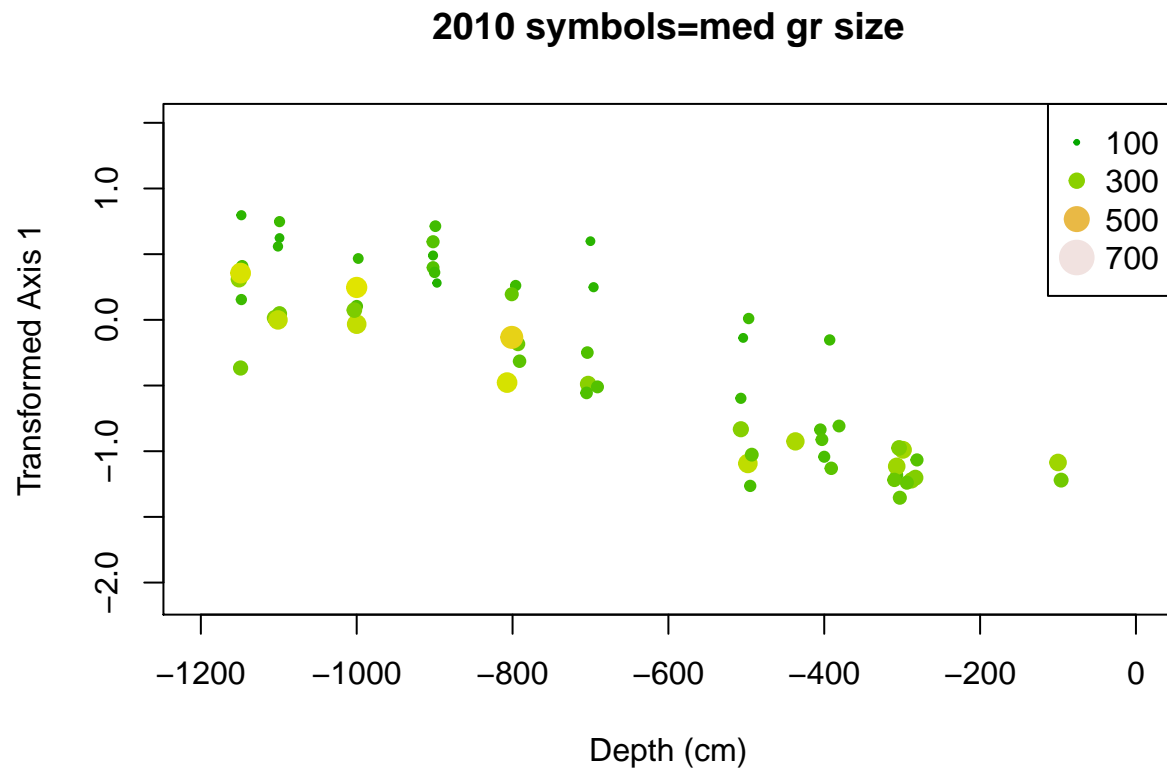


Figure 5: prediction based on all axis 1 scores

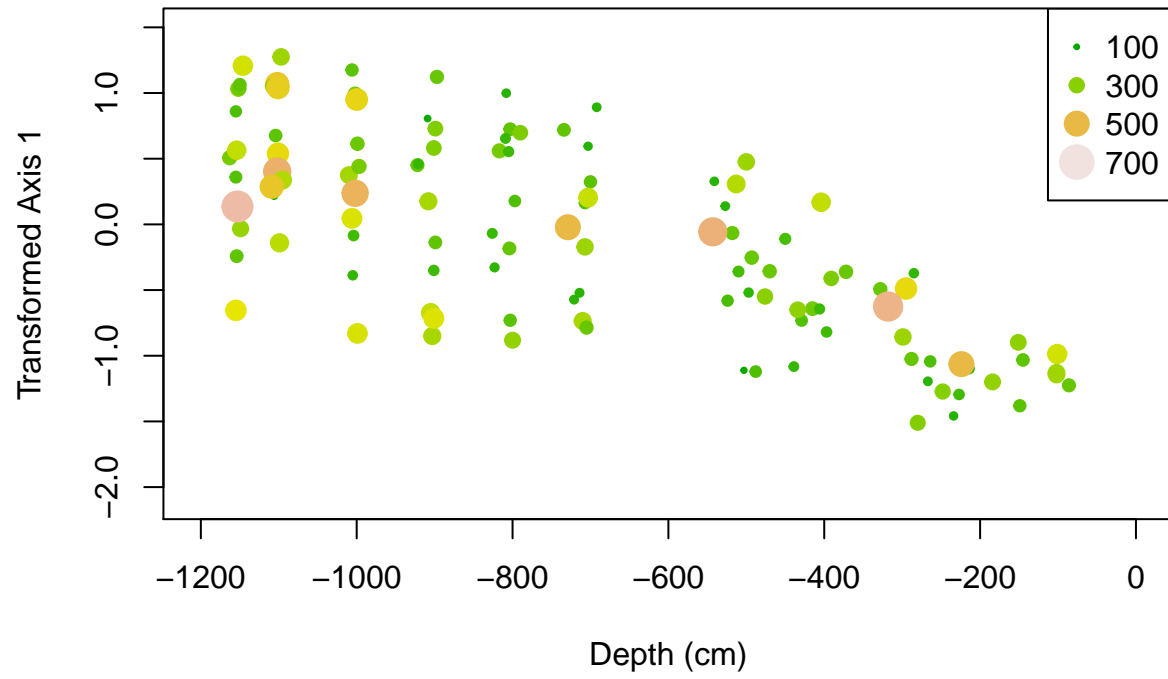
```

legend("topright",legend=c("100","300","500","700"),
      col=terrain.colors(256)[(c(100,300,500,700)-90)/630*256],
      pt.cex=c(100,300,500,700)/300,pch=19)
}

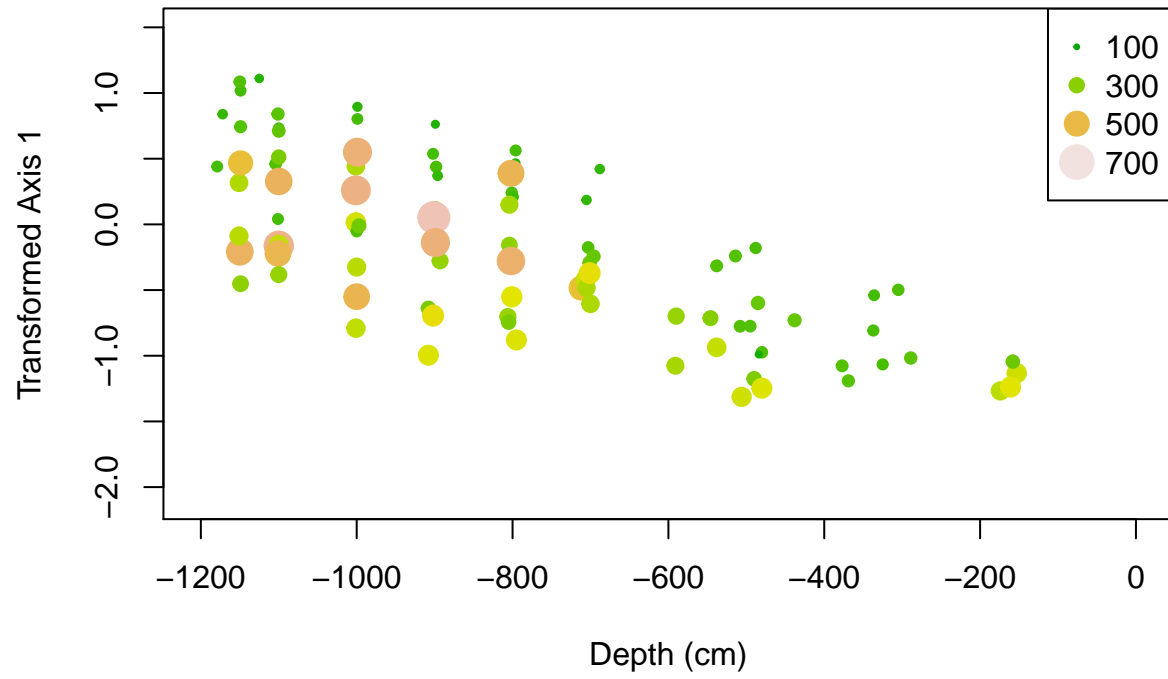
```



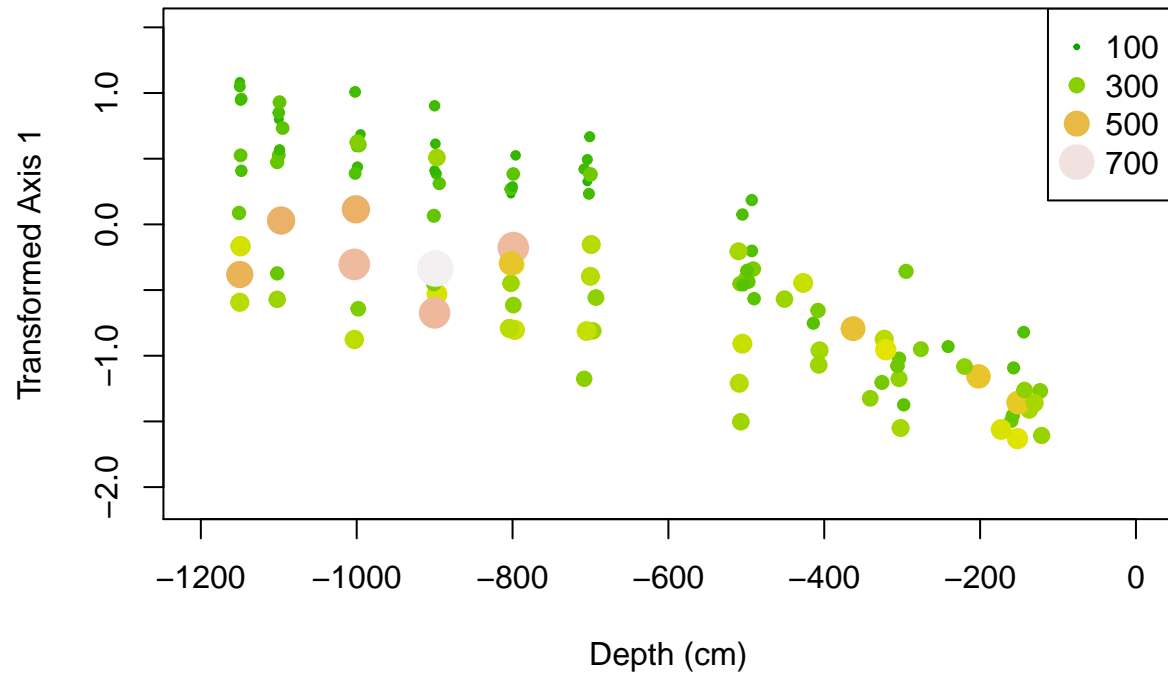
2012 symbols=med gr size



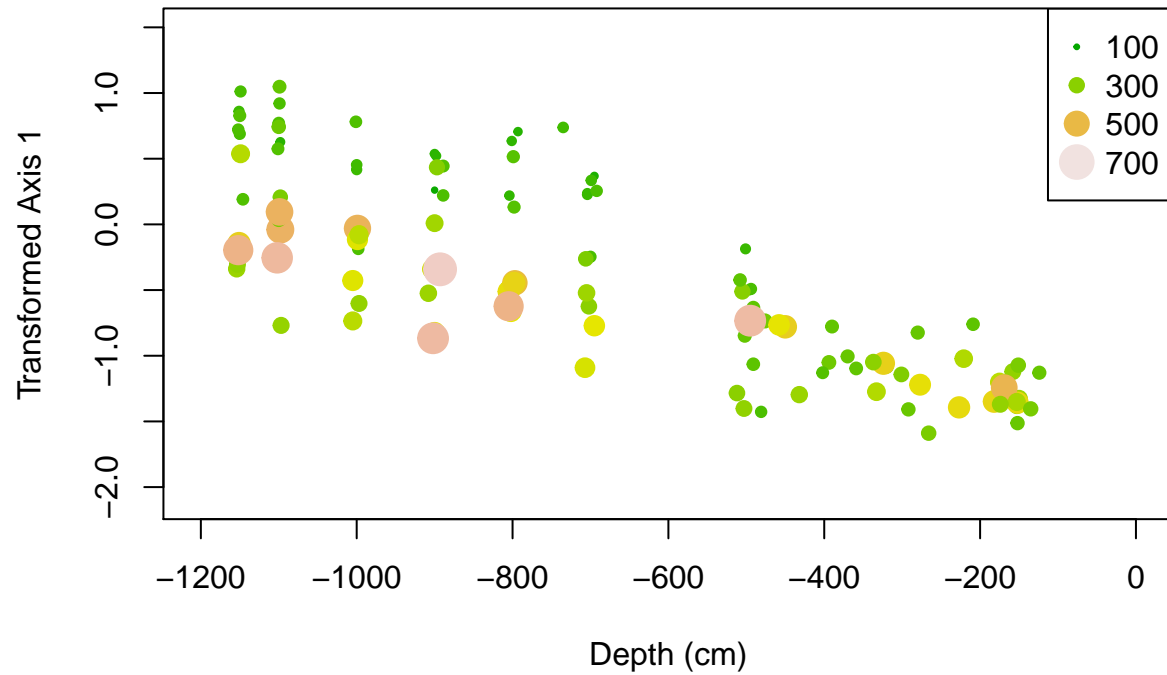
2013 symbols=med gr size



2015 symbols=med gr size

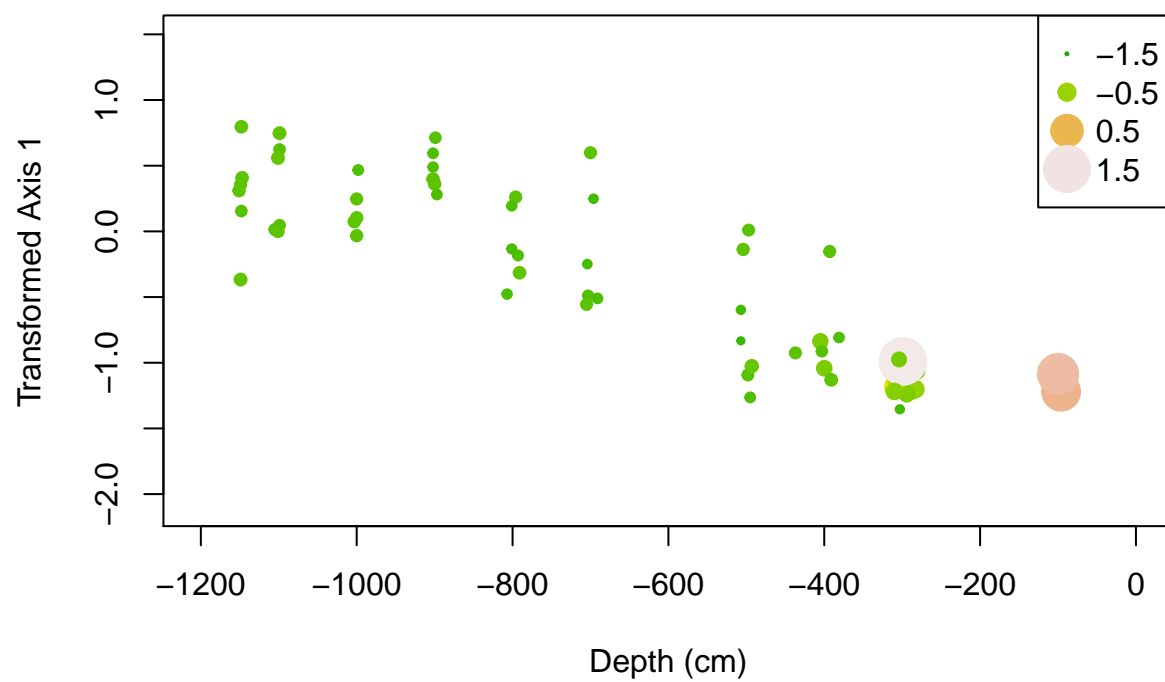


2017 symbols=med gr size

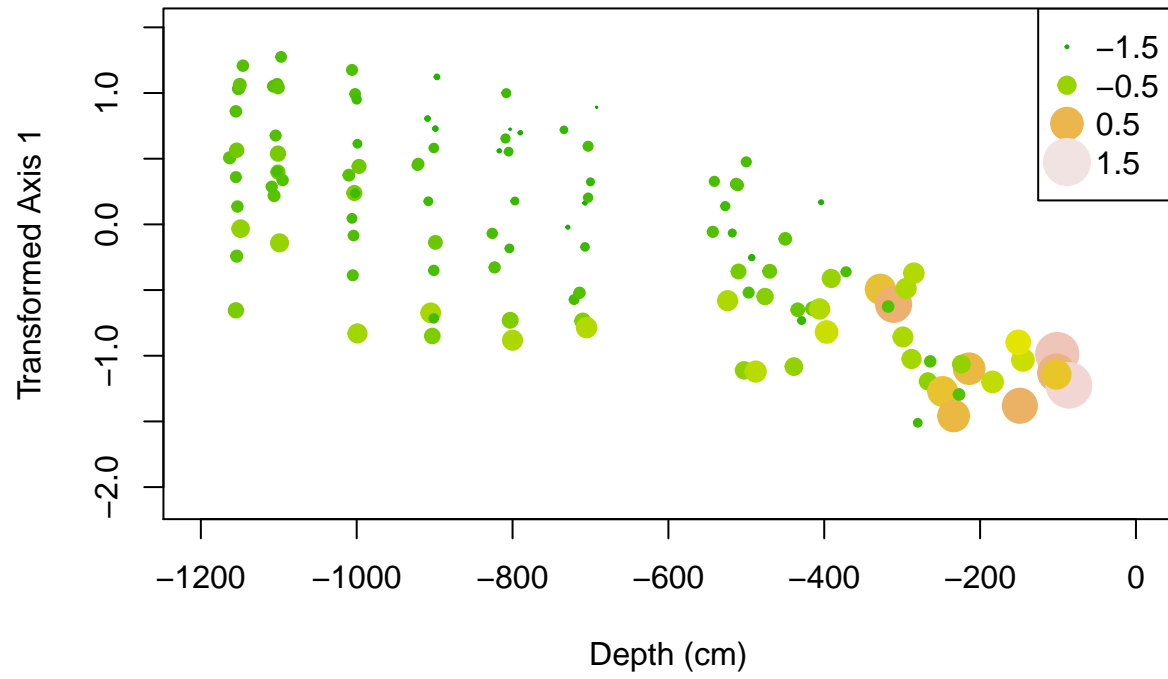


```
for (jr in c(2010,2012,2013,2015,2017)){
  symb<-log(smpca$tt[smpca$jaar==jr])
  cls<-terrain.colors(256)[(symb+1.7)/3.3*256]
  plot(smpca$dep_cm[smpca$jaar==jr],smpca$a1t[smpca$jaar==jr],
       cex=symb+1.7,col=cls,pch=19,
       main=paste(jr,"symbols=log(tautot)"),
       xlab="Depth (cm)",ylab="Transformed Axis 1",
       xlim=c(-1200,0),ylim=c(-2.1,1.5))
  legend("topright",legend=c("-1.5","-0.5","0.5","1.5"),
        col=terrain.colors(256)[(c(-1.5,-.5,.5,1.5)+1.7)/3.3*256],
        pt.cex=c(-1.5,-.5,.5,1.5)+1.7,pch=19)
}
```

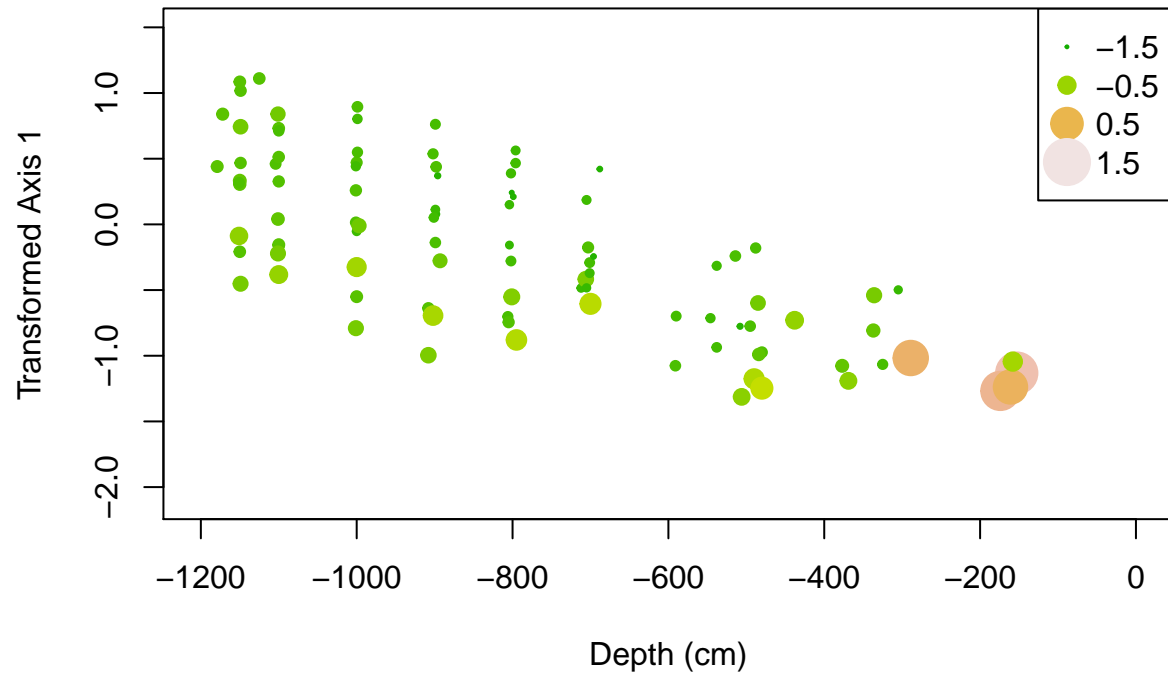
2010 symbols=log(tautot)



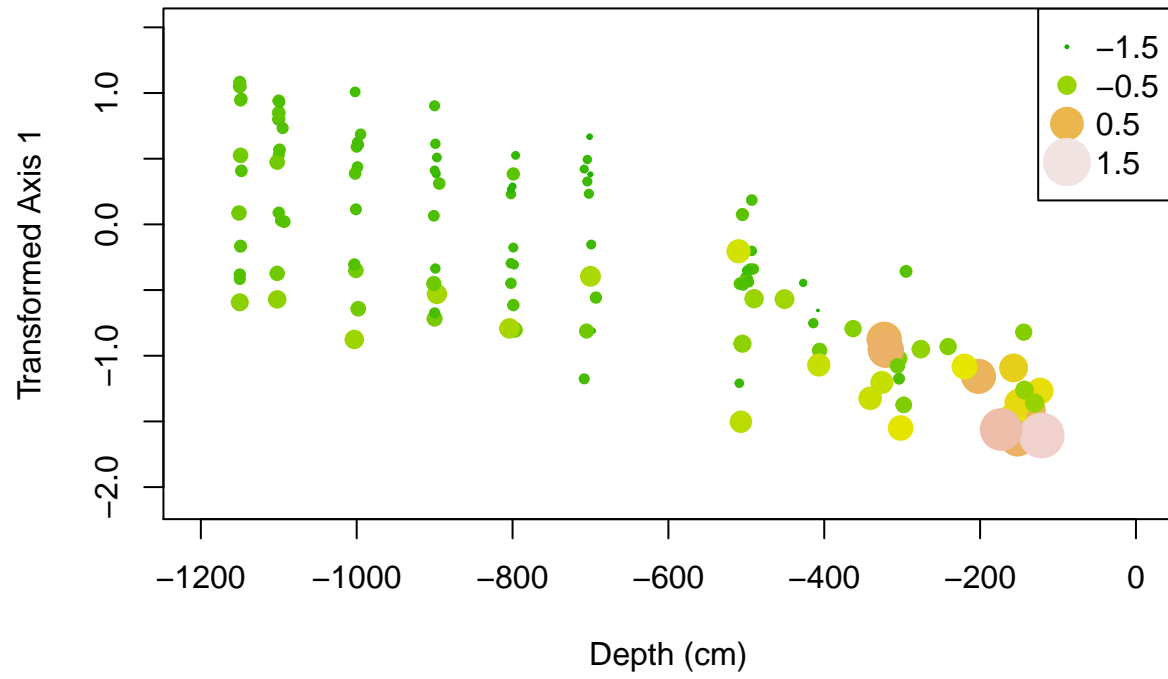
2012 symbols=log(tautot)



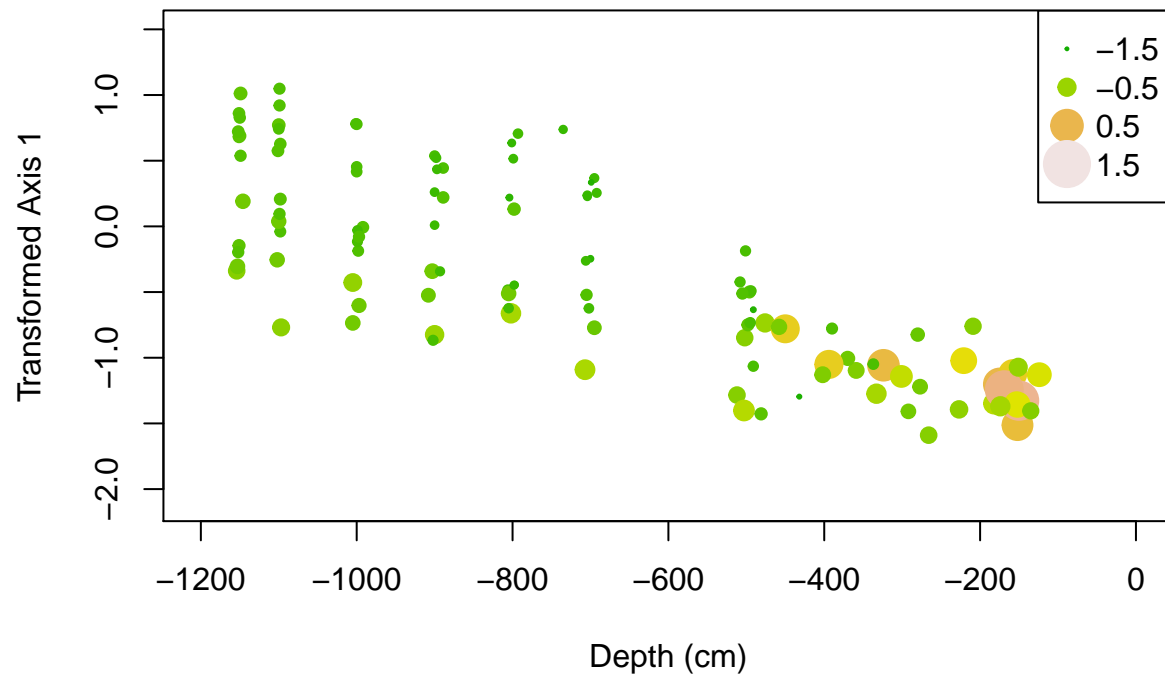
2013 symbols=log(tautot)



2015 symbols=log(tautot)



2017 symbols=log(tautot)



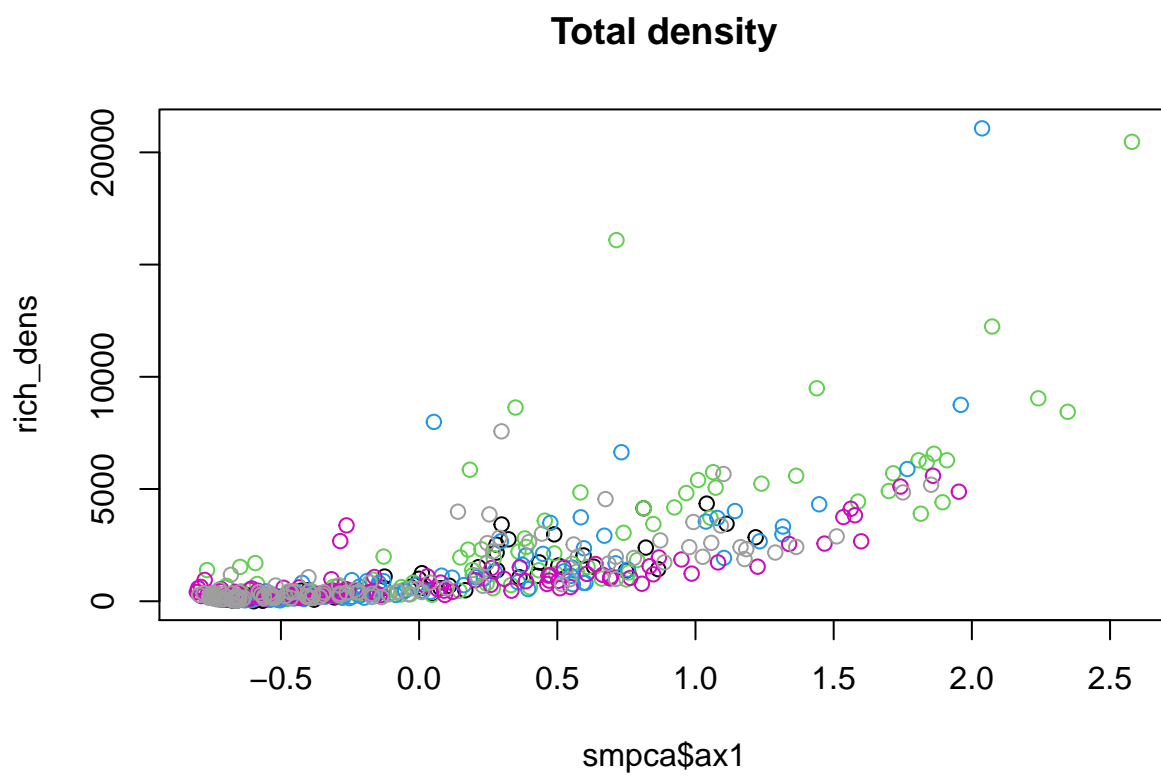
Analysis of the species in relation to the pca ordination

In this analysis we investigate how species score on the ordination axis (especially the first axis), what their abundance is and how they relate to the environmental factors. We also investigate how representative the first axis is for the richness of the community. We employ total density, sum of log(density) and number of species as measures of richness, thereby assuming that sum of log(density) may be the best indicator.

We also added a plot of the spatial distribution of the most frequent species (freq>30) in the different years. In a previous step of the analysis, we used color-coding according to laboratory, to check for any suspect patterns.

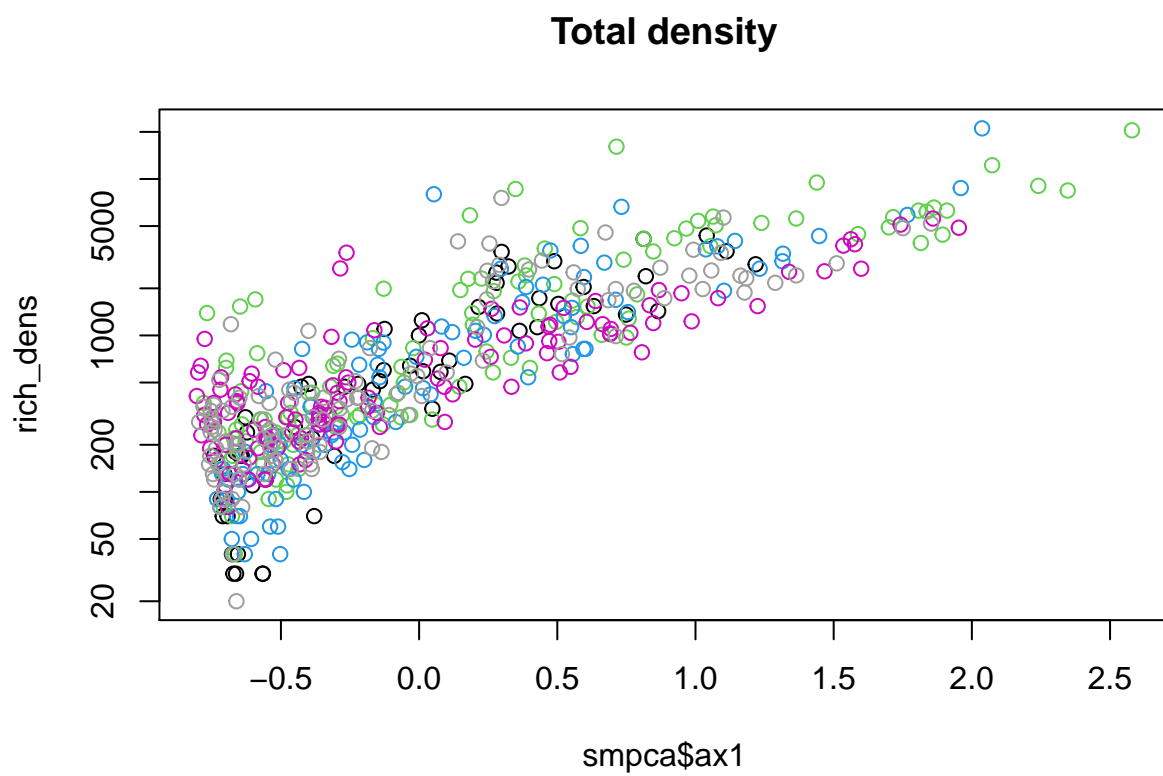
```
# first investigate the relation between richness and scores on the first axis
rich_dens<-apply(dens,1,sum) # total density in the sample
rich_ldens<-apply(ldens,1,sum) # sum of log(density+1)
rich_nspec<-apply(dens,1,FUN=function(x)length(x[x>0]))

plot(smpca$ax1,rich_dens,col=smpca$jaar-2009,main="Total density")
```

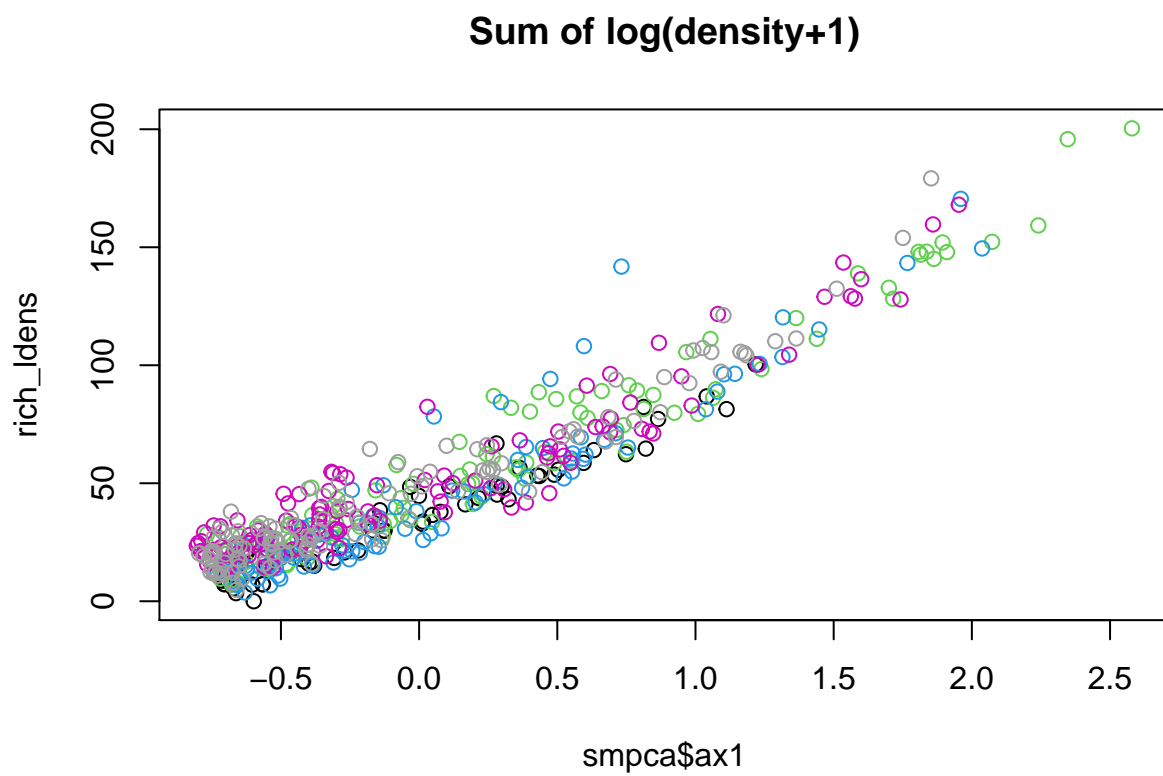


```
plot(smPCA$ax1,rich_dens,col=smPCA$jaar-2009,main="Total density",log="y")
```

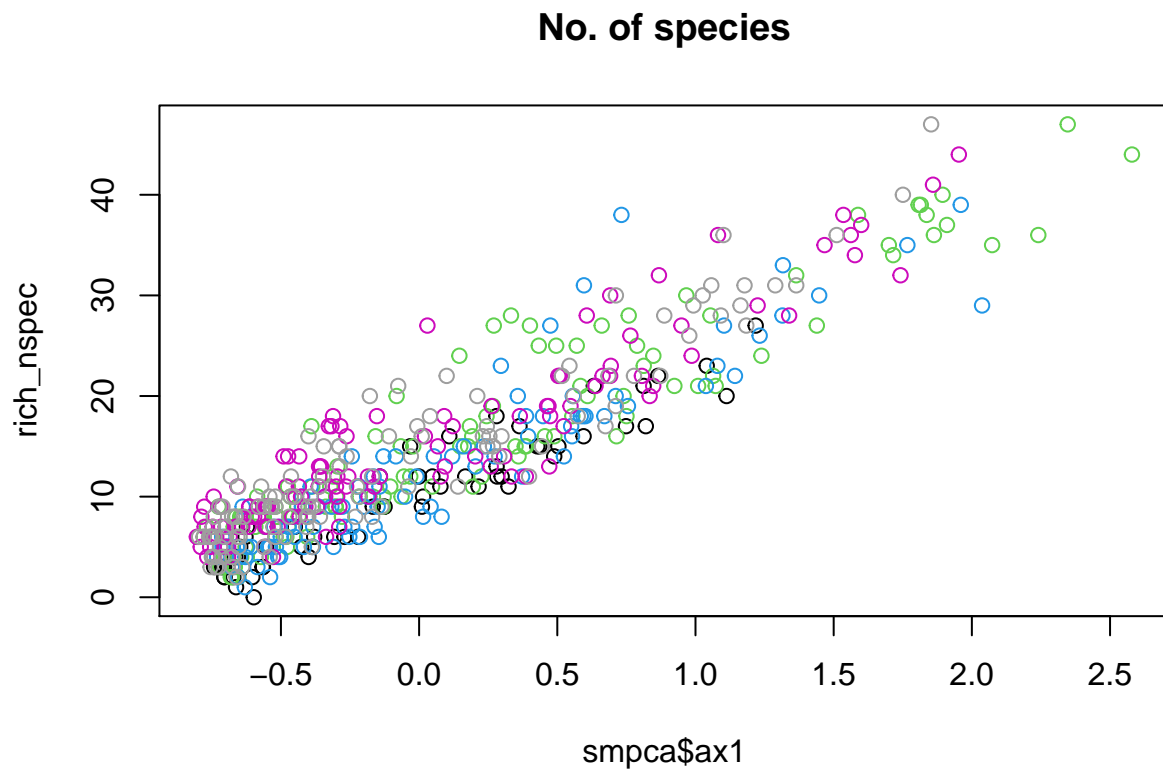
```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 y value <= 0 omitted from  
## logarithmic plot
```



```
plot(smpca$ax1,rich_ldens,col=smpca$jaar-2009,main= "Sum of log(density+1)")
```

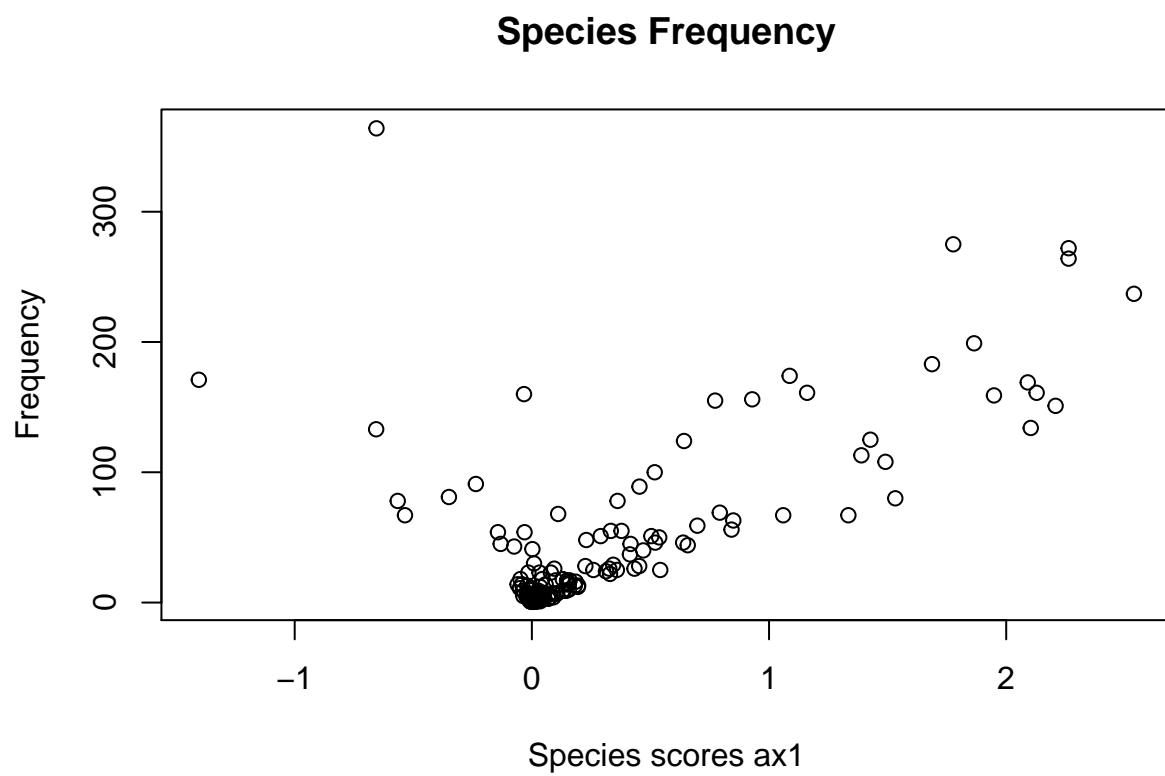


```
plot(smpca$ax1,rich_nspec,col=smpca$jaar-2009,main= "No. of species")
```



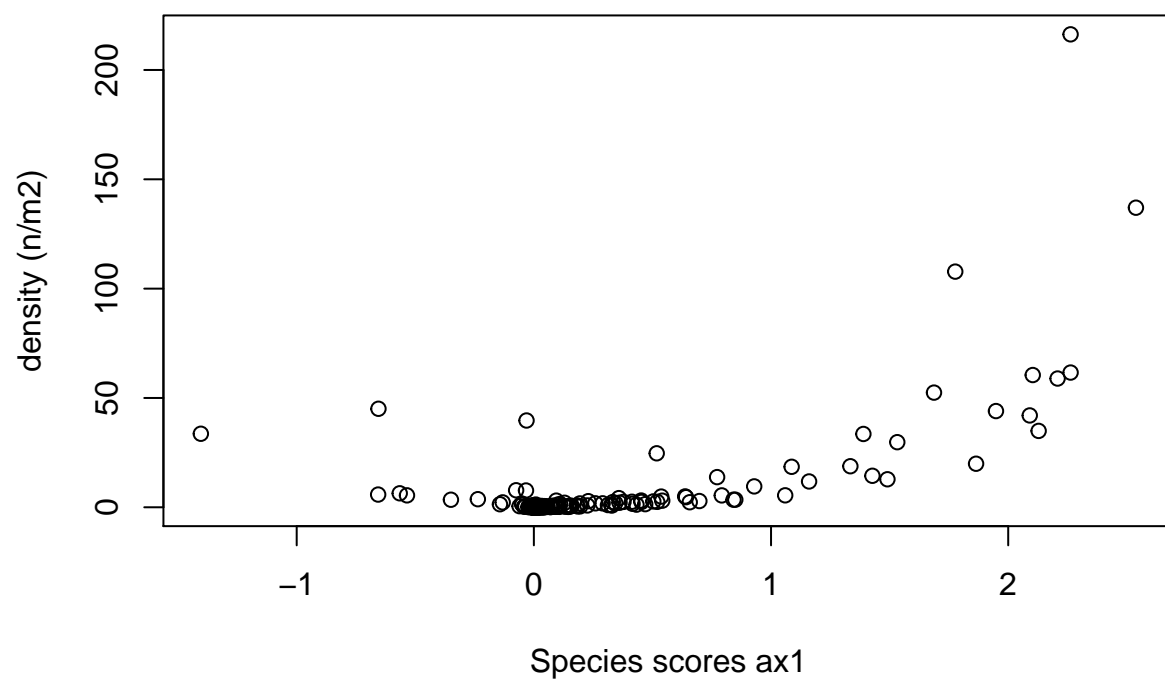
```
# next we calculate for each species how often it has been found,
# and what its average density and ldens over all stations was
sp_freq  <- apply(dens,2,FUN=function(x)length(x[x>0]))
sp_mdens <- apply(dens,2,mean)
sp_mldens <- apply(ldens,2,mean)

plot(scores(pca)$species[,1],sp_freq,main="Species Frequency",
      xlab="Species scores ax1",ylab="Frequency")
```

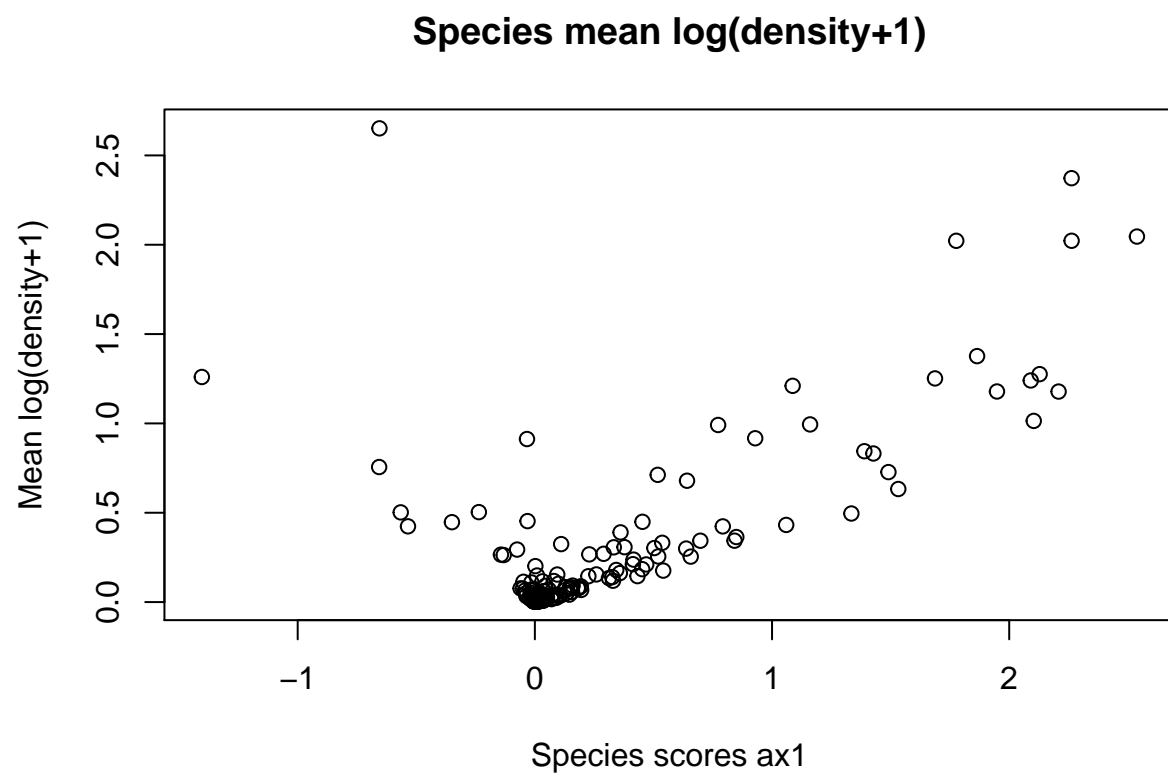



```
plot(scores(pca)$species[,1],sp_mdens,main="Species mean density",  
      xlab="Species scores ax1",ylab="density (n/m2)")
```

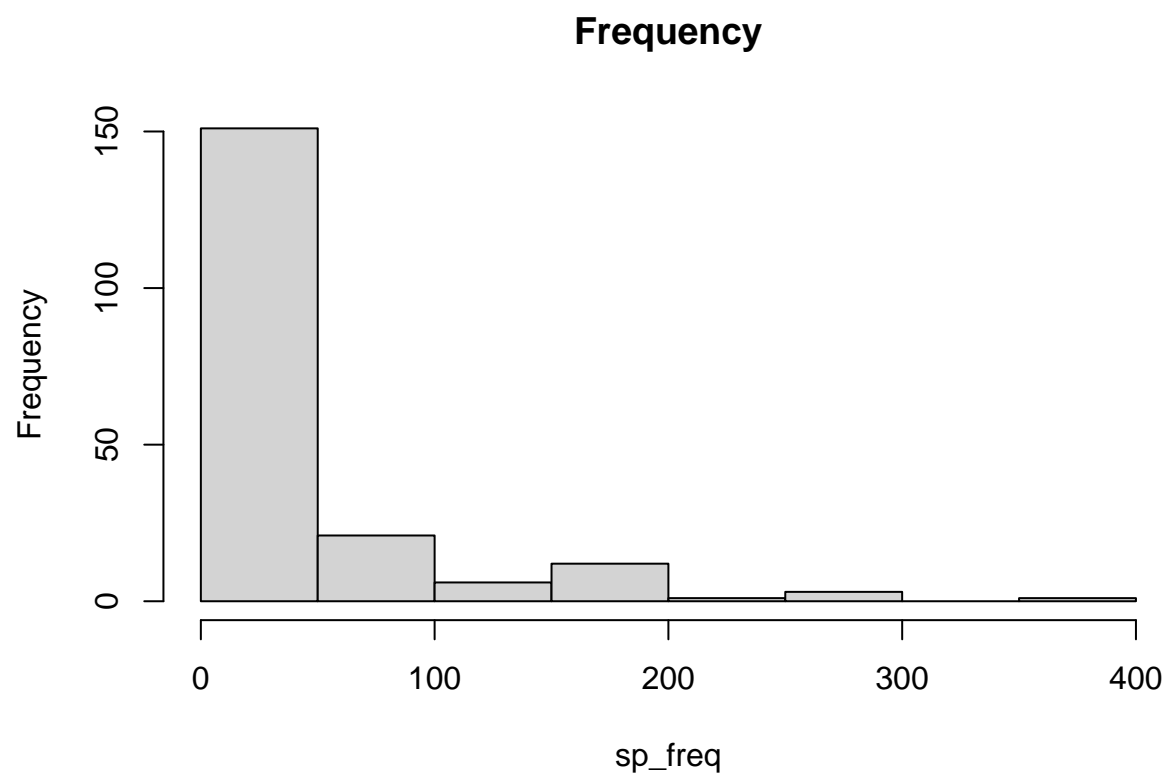
Species mean density



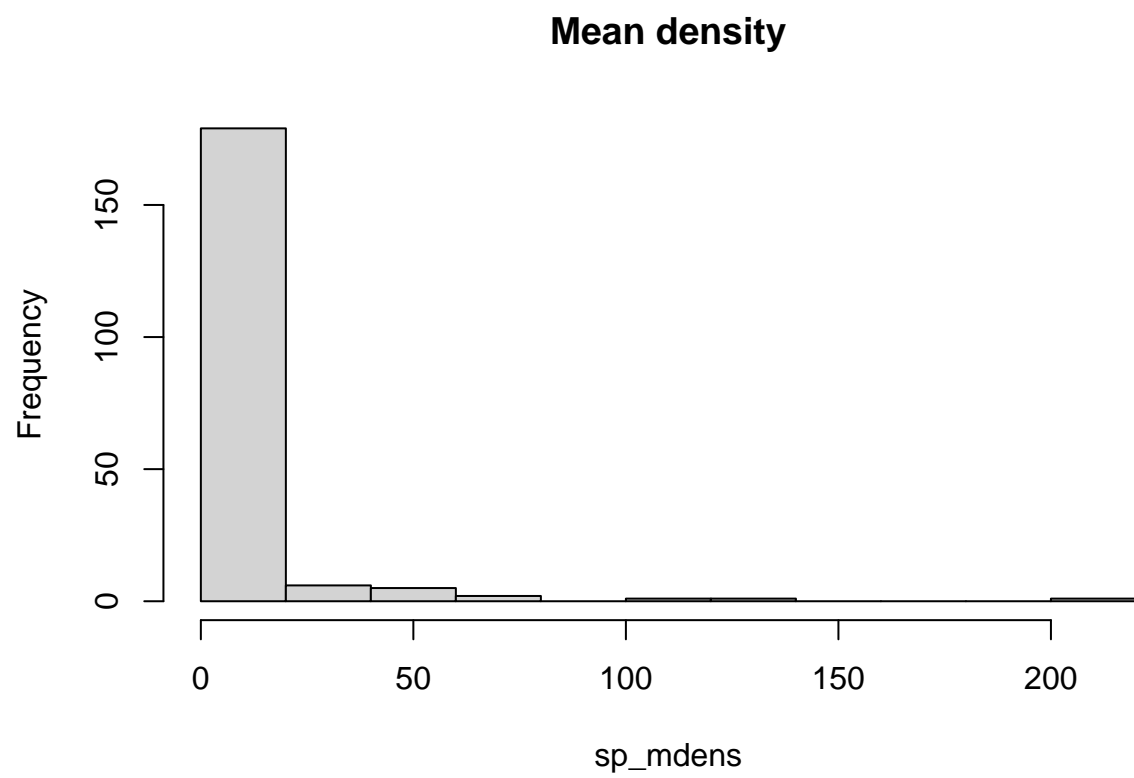
```
plot(scores(pca)$species[,1],sp_mldens,main="Species mean log(density+1)",  
      xlab="Species scores ax1",ylab="Mean log(density+1)")
```



```
# plot histograms of species characteristics  
hist(sp_freq,main="Frequency")
```

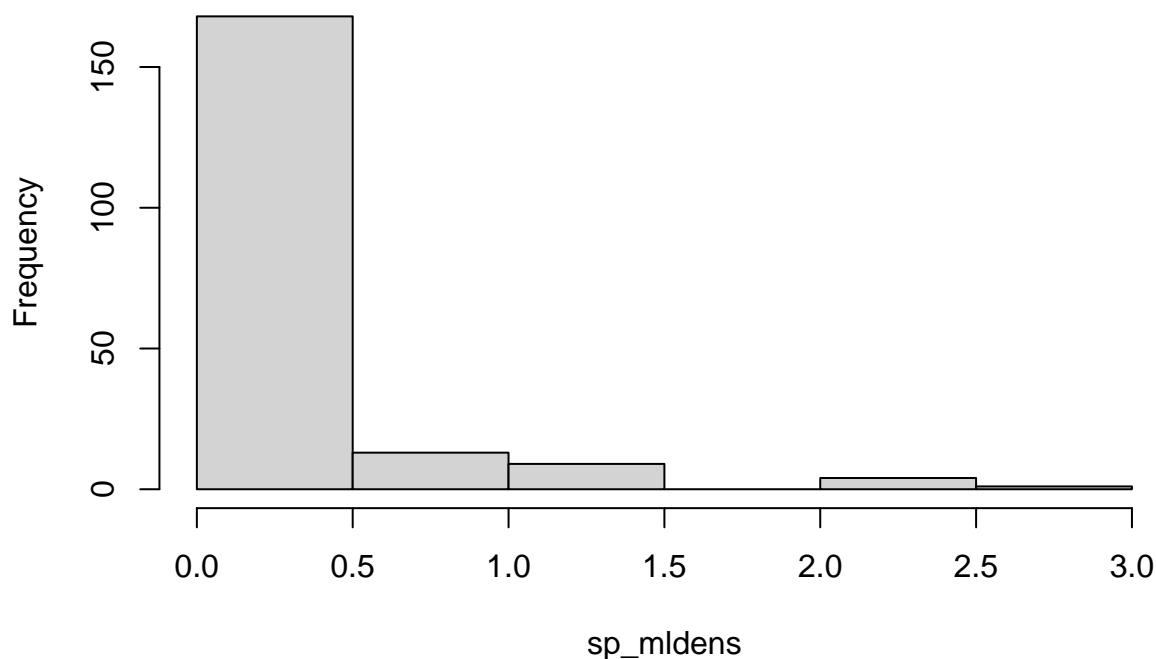


```
hist(sp_mdens,main="Mean density")
```



```
hist(sp_mdens,main="Mean log(density+1)")
```

Mean log(density+1)



```
sp_pca<-data.frame(ax1=scores(pca)$species[,1],
                   ax2=scores(pca)$species[,2],
                   freq=sp_freq,mdens=sp_mdens,mldens=sp_mldens)
sp_pca<-sp_pca[order(sp_pca$ax1),]
fr_sp_pca<-sp_pca[sp_pca$freq>30,]
print(fr_sp_pca,digits=3)
```

##		ax1	ax2	freq	mdens	mldens
##	Scoelepis (Scoelepis) squamata	-1.40408	1.16753	171	33.62	1.260
##	Gastrosaccus spinifer	-0.65650	0.28815	133	5.82	0.756
##	Nephtys cirrosa	-0.65518	-1.35467	364	45.06	2.651
##	Paraonis fulgens	-0.56552	0.35544	78	6.40	0.502
##	Haustorius arenarius	-0.53496	0.51881	67	5.45	0.424
##	Portumnus latipes	-0.34962	0.05046	81	3.45	0.447
##	Bathyporeia elegans	-0.23613	-0.23554	91	3.70	0.503
##	Pontocrates altamarinus	-0.14301	0.02669	54	1.39	0.265
##	Pontocrates arenarius	-0.13144	-0.01885	45	2.24	0.263
##	Ophelia	-0.07474	0.06151	43	7.78	0.294
##	Diogenes pugilator	-0.03315	-0.74915	160	7.67	0.913
##	Microphthalmus similis	-0.03088	0.10480	54	39.69	0.453
##	Schistomysis kervillei	0.00214	-0.11505	41	1.00	0.201
##	Hydrozoa	0.11084	-0.01289	68	1.50	0.324
##	Glycera	0.22950	0.07250	48	2.79	0.267
##	Liocarcinus holsatus	0.29005	0.03363	51	1.81	0.270
##	Euspira nitida	0.33280	0.19443	55	2.38	0.307
##	Nototropis falcatus	0.36129	-0.09334	78	2.08	0.390

## <i>Scolecopsis bonnieri</i>	0.37793	-0.11527	55	2.47	0.307
## <i>Echinocardium cordatum</i>	0.41349	0.12918	37	2.52	0.212
## <i>Nototropis swammerdamei</i>	0.41605	0.11567	45	1.60	0.238
## <i>Crangon crangon</i>	0.45373	0.00376	89	2.45	0.449
## <i>Phyllodoce (Anaitides) groenlandica</i>	0.46942	0.12577	40	1.43	0.211
## <i>Urothoe poseidonis</i>	0.50369	0.15303	51	2.68	0.302
## <i>Donax vittatus</i>	0.51796	-0.48457	100	24.68	0.712
## <i>Ophiuroidea</i>	0.52018	0.17657	46	2.49	0.255
## <i>Oligochaeta</i>	0.53699	0.34622	50	4.89	0.332
## <i>Malmgrenia</i>	0.63797	0.15169	46	4.97	0.299
## <i>Diastylis bradyi</i>	0.64158	0.15793	124	4.62	0.679
## <i>Poecilochaetus serpens</i>	0.65718	0.16470	44	2.31	0.254
## <i>Leucothoe incisa</i>	0.69781	0.23081	59	2.85	0.343
## <i>Magelona mirabilis</i>	0.77270	-0.85141	155	13.80	0.991
## <i>Malmgrenia darbouxi</i>	0.79233	0.37603	69	5.43	0.424
## <i>Lagis koreni</i>	0.84185	0.34449	56	3.47	0.344
## <i>Caprellidae</i>	0.84933	0.42723	63	3.47	0.364
## <i>Nemertea</i>	0.92906	-0.10829	156	9.56	0.917
## <i>Notomastus latericeus</i>	1.05966	0.46202	67	5.47	0.432
## <i>Nephtys</i>	1.08690	0.11705	174	18.52	1.210
## <i>Eteone</i>	1.16041	-0.26163	161	11.81	0.994
## <i>Eumida</i>	1.33454	0.51109	67	18.77	0.496
## <i>Macoma balthica</i>	1.38936	0.15227	113	33.53	0.844
## <i>Phyllodoce mucosa</i>	1.42732	-0.07805	125	14.41	0.832
## <i>Kurtiella bidentata</i>	1.49080	0.67514	108	12.81	0.727
## <i>Owenia</i>	1.53232	0.99304	80	29.77	0.633
## <i>Capitella</i>	1.68706	-0.16637	183	52.45	1.252
## <i>Spio</i>	1.77668	-1.31776	275	107.78	2.022
## <i>Nephtys hombergii</i>	1.86447	-0.33902	199	19.92	1.377
## <i>Spisula subtruncata</i>	1.94844	0.45131	159	44.01	1.179
## <i>Scoloplos armiger</i>	2.09084	0.71949	169	42.00	1.240
## <i>Abra alba</i>	2.10346	0.72082	134	60.48	1.014
## <i>Fabulina fabula</i>	2.12823	0.15227	161	34.93	1.276
## <i>Lanice conchilega</i>	2.20817	0.65872	151	58.88	1.178
## <i>Ensis</i>	2.26295	-1.12697	264	216.29	2.372
## <i>Spiophanes bombyx</i>	2.26303	-0.31897	272	61.62	2.022
## <i>Magelona johnstoni</i>	2.53859	-1.25082	237	137.01	2.046

```
# plot some species' occurrence over the years
```

```
# function to plot a species
```

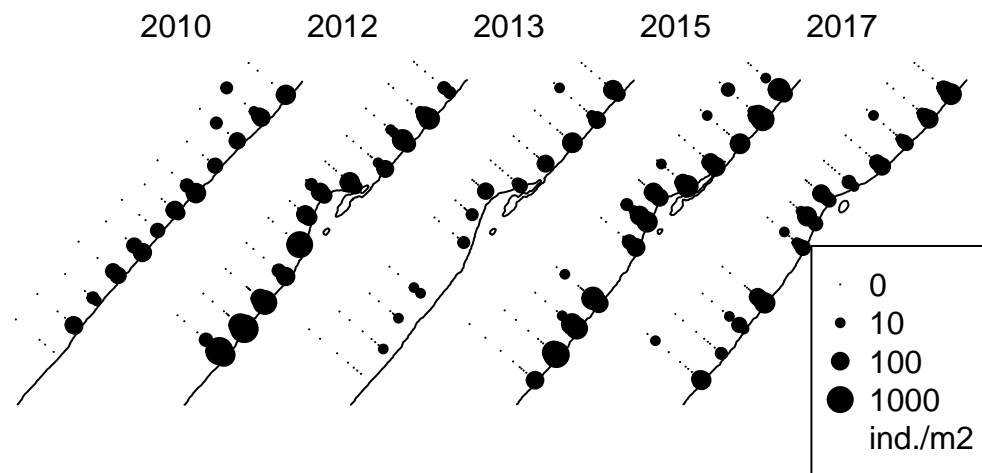
```
sp_plot<-function(nam,densval,cexf=5){
  plot(coordinates(smpca)[,1]+as.numeric(smpca$jaarf)*5000,coordinates(smpca)[,2],
        pch=19,cex=.01+densval/cexf,asp=1,axes=F,xlab="",ylab="",main=nam)
  text(seq(78000,98000,by=5000),rep(458000,5),c("2010","2012","2013","2015","2017"))
  #legend("topleft",legend=levels(smpca$lab),col=1:7,pch=19)
}
for(i in 1:nrow(fr_sp_pca)){
  nam<-row.names(fr_sp_pca)[i]
  sp_plot(nam,ldens[,nam],4)
  nams<-c("bath_2010","bath_2012","bath_2013","bath_2015","bath_2017")
  for(j in 1:5){
    bb<-subset(b,nams[j])
    bb<-shift(bb,5000*j)
```

```

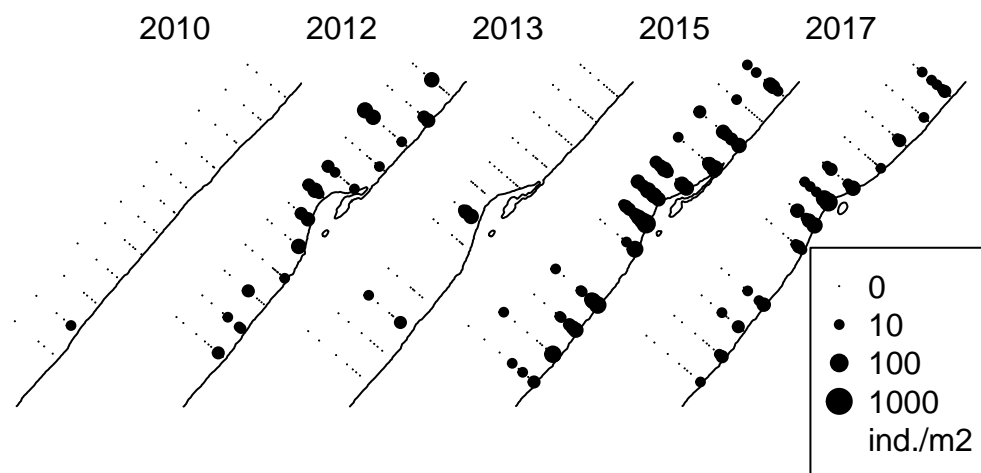
a<-rasterToContour(bb,levels=c(0))
plot(a,add=T)
}
legend("bottomright",legend=c("0","10","100","1000","ind./m2"),
      pt.cex=0.01+log(c(0,10,100,1000)+1)/4,pch=c(rep(19,4),NA))
}

```

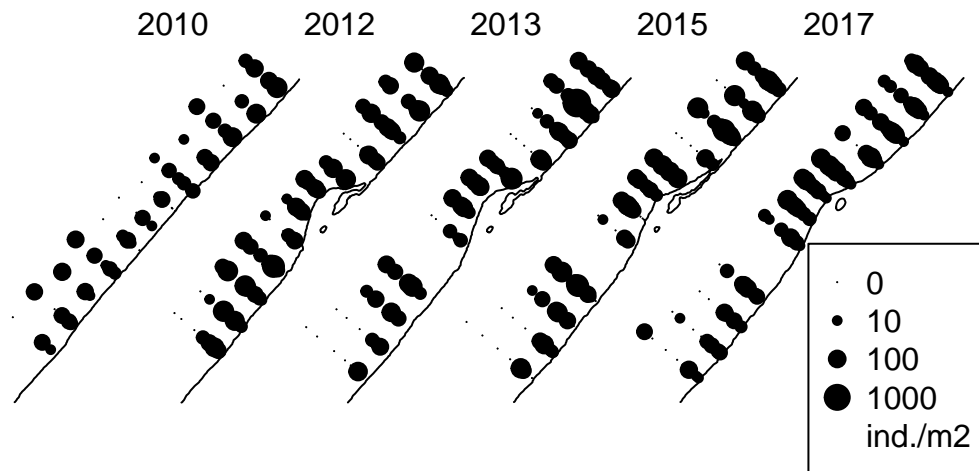
Scolecopsis (Scolecopsis) squamata



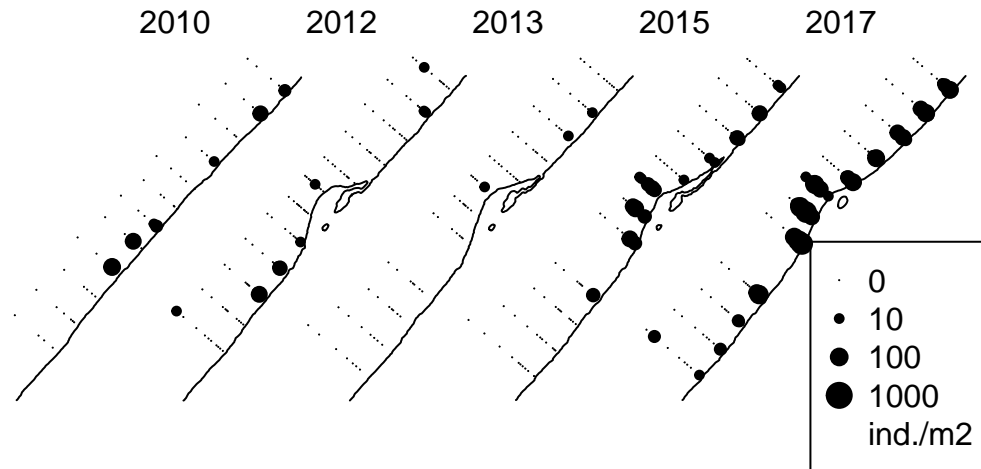
Gastrosaccus spinifer



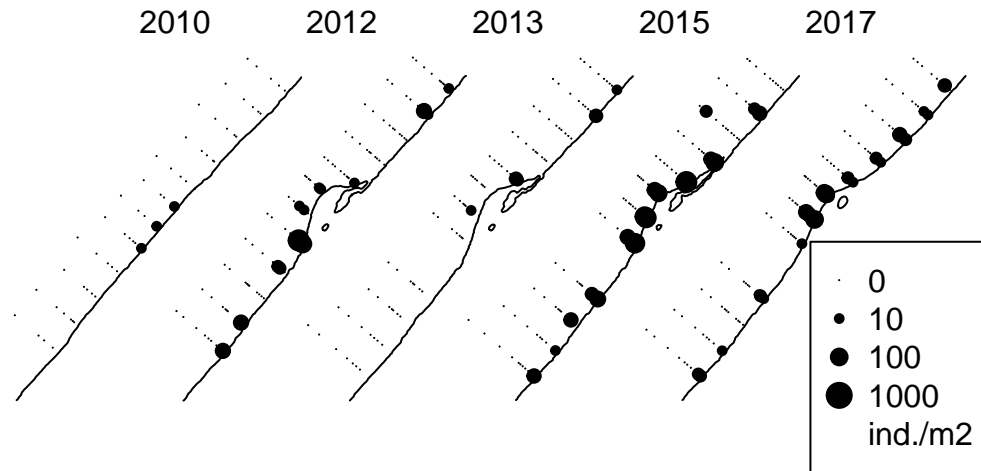
Nephtys cirrosa



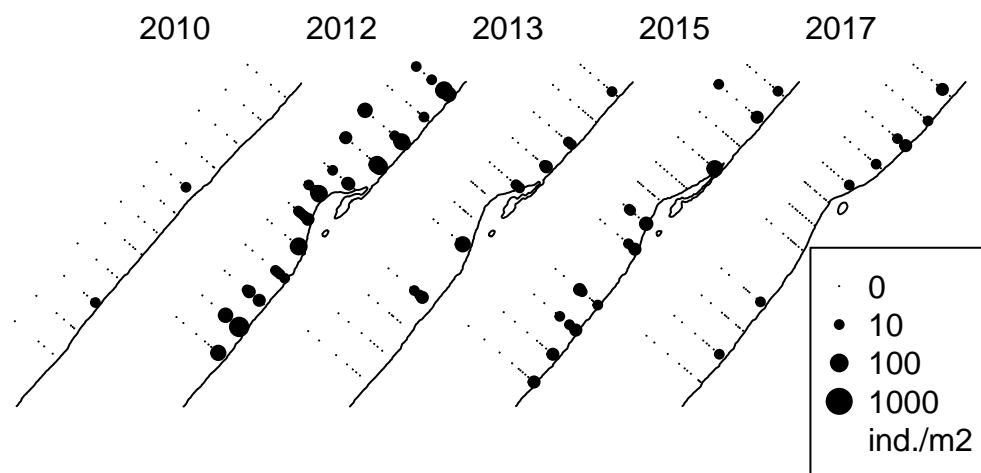
Paraonis fulgens



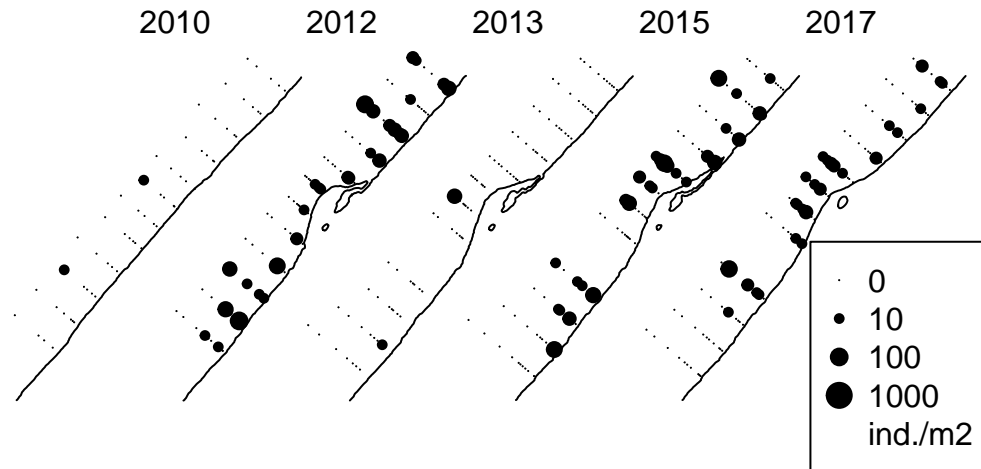
Haustorius arenarius



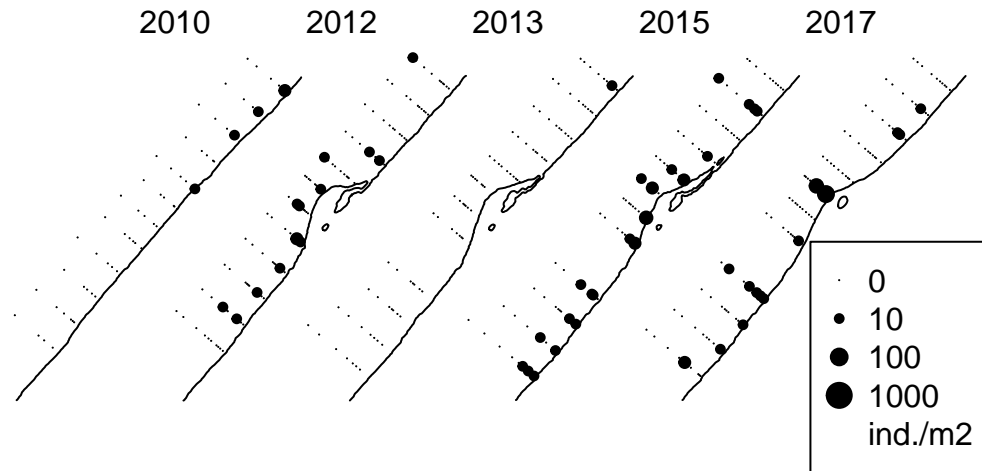
Portumnus latipes



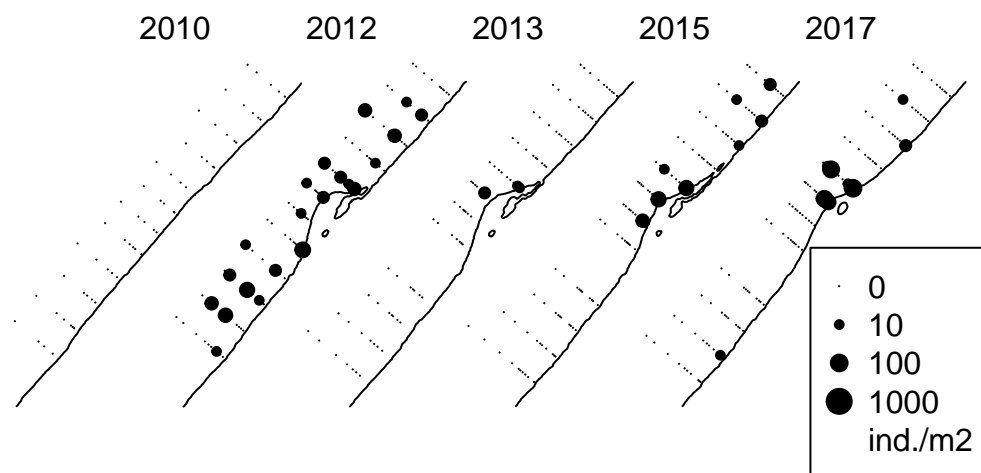
Bathyporeia elegans



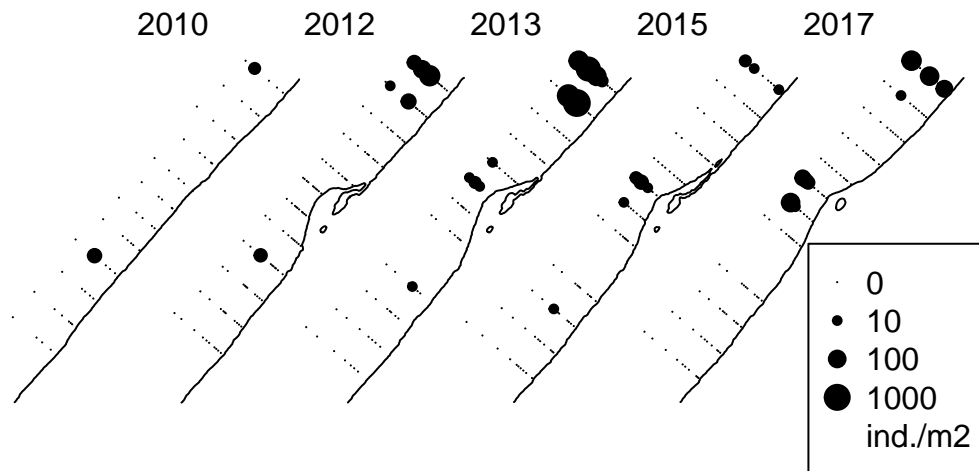
Pontocrates altamarinus



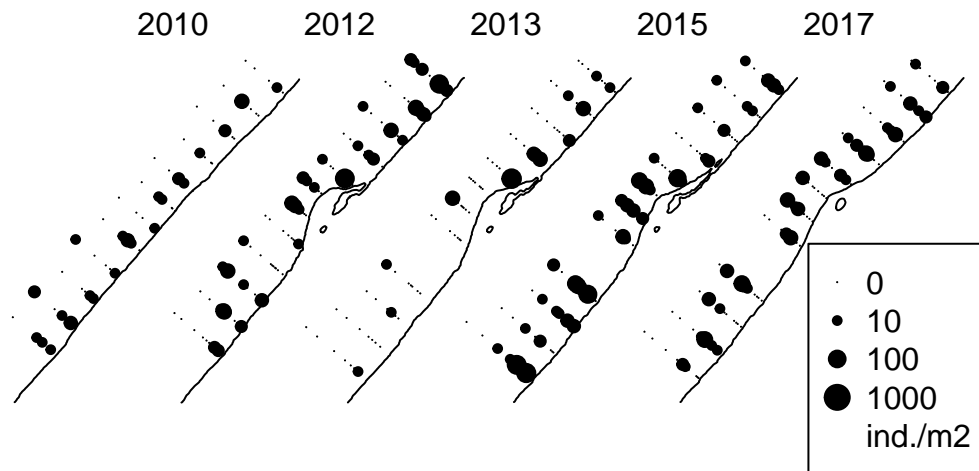
Pontocrates arenarius



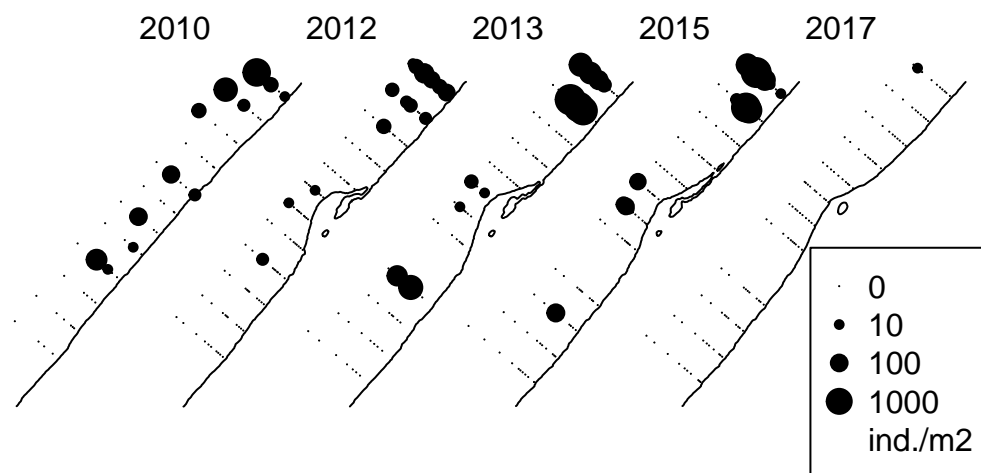
Ophelia



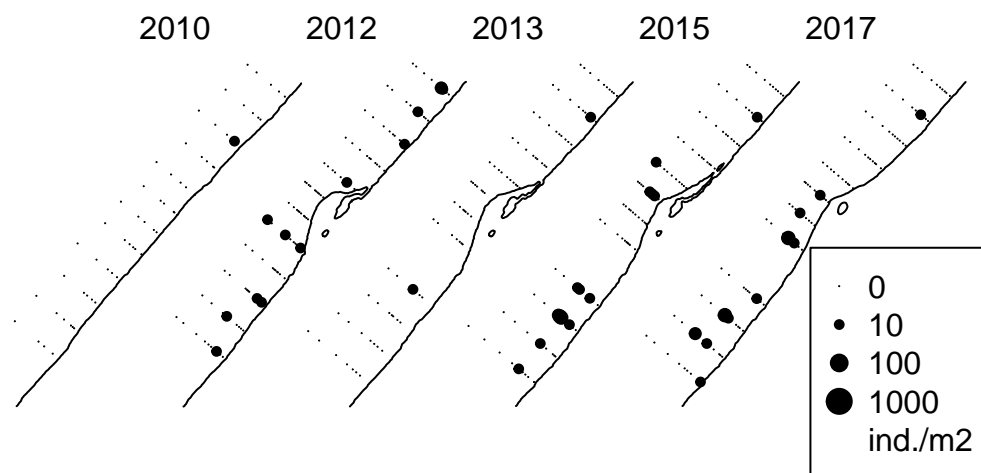
Diogenes pugilator



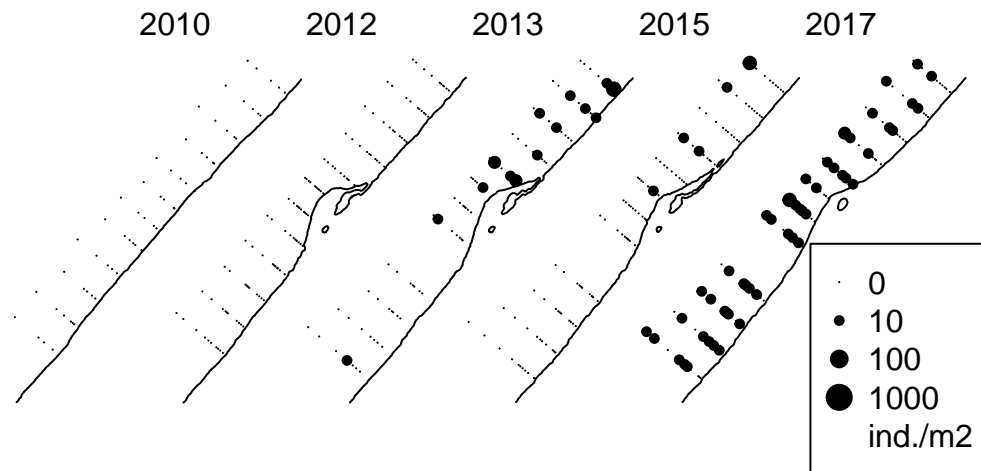
Microphthalmus similis



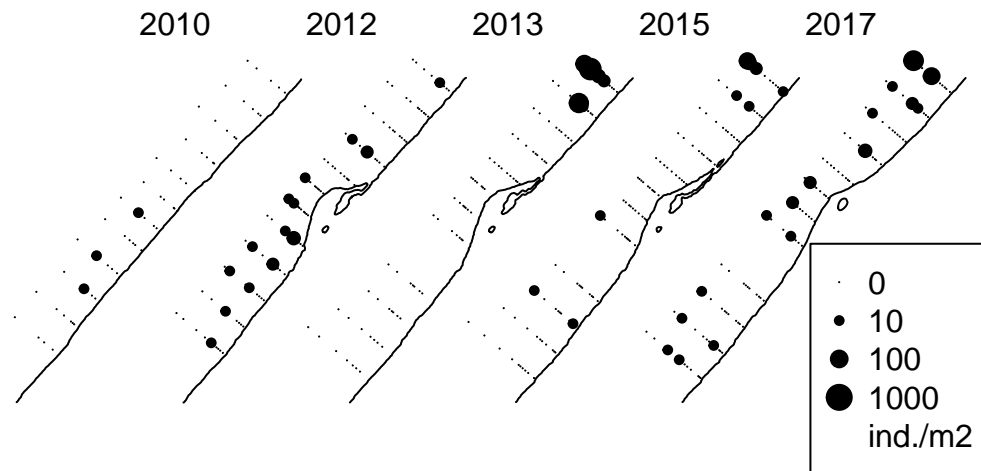
Schistomysis kervillei



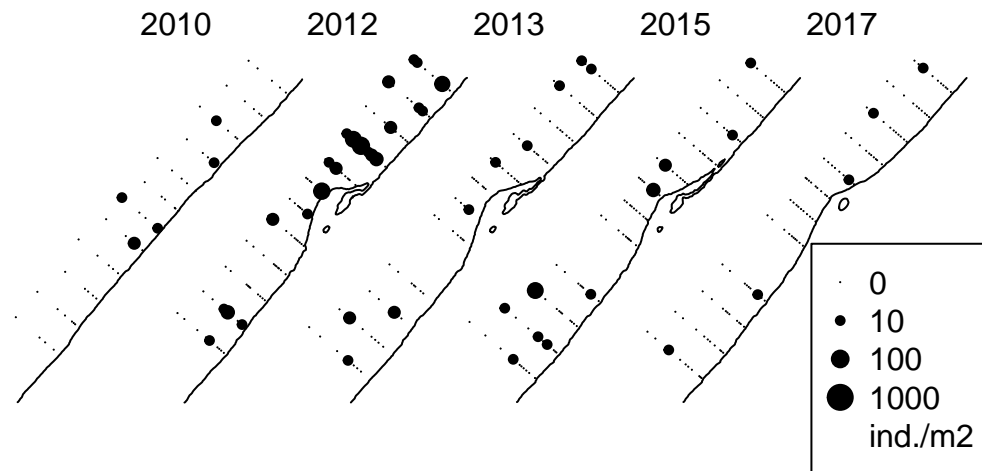
Hydrozoa



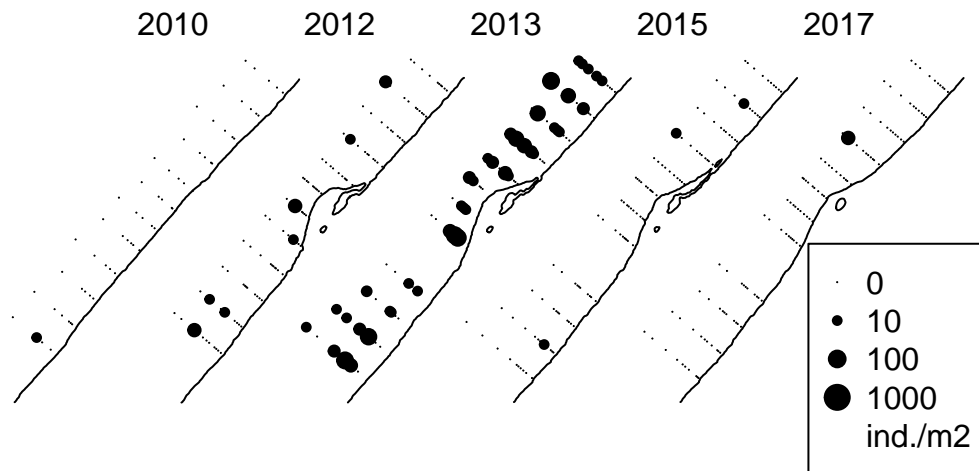
Glycera



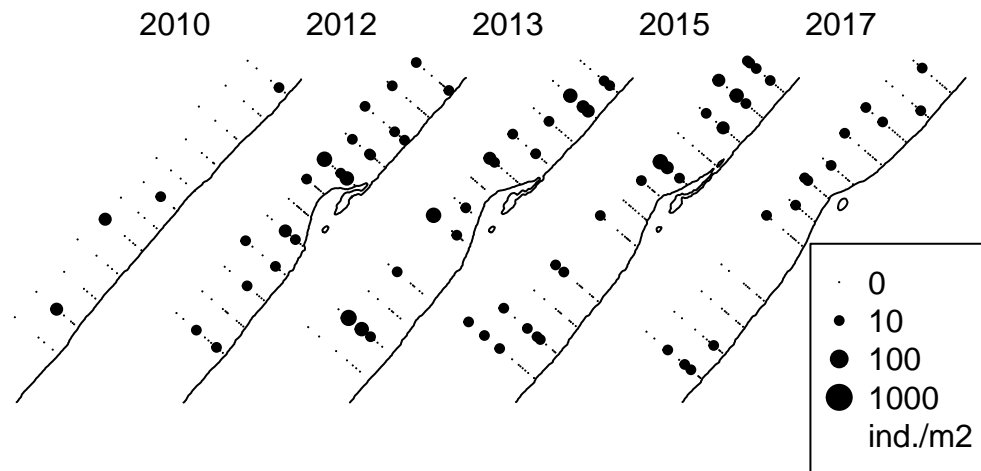
Liocarcinus holsatus



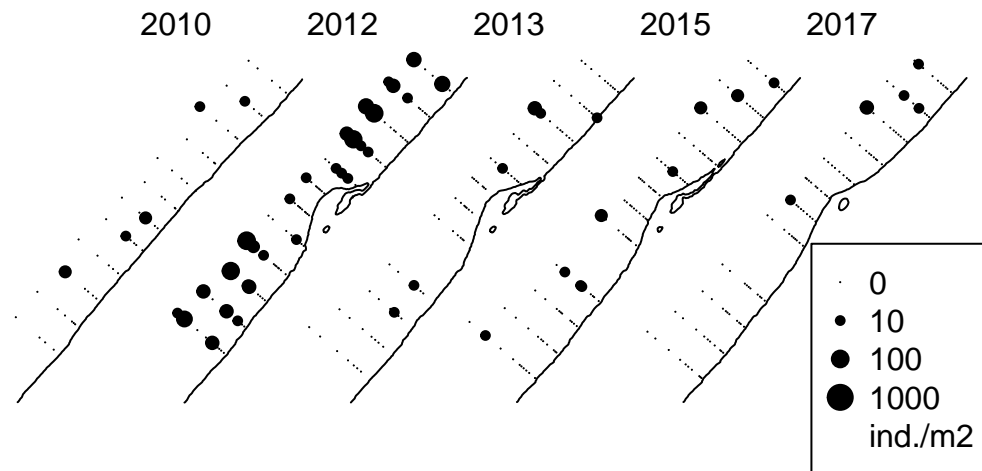
Euspira nitida



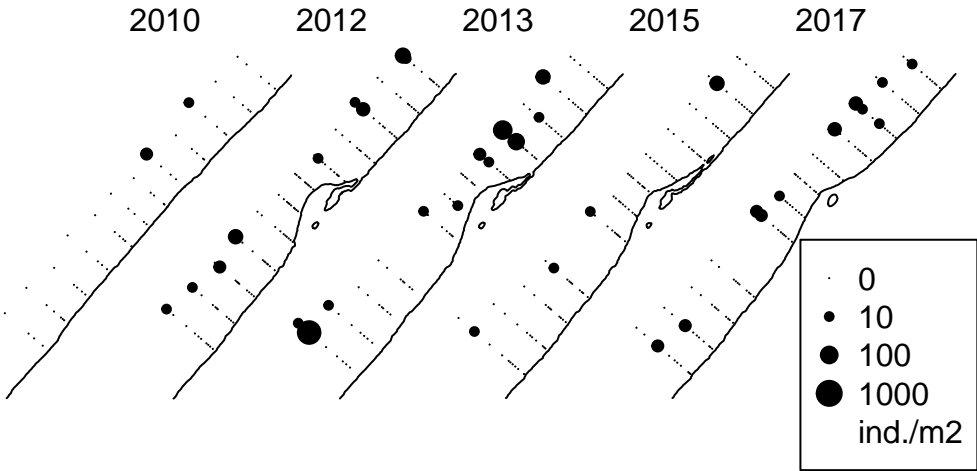
Nototropis falcatus



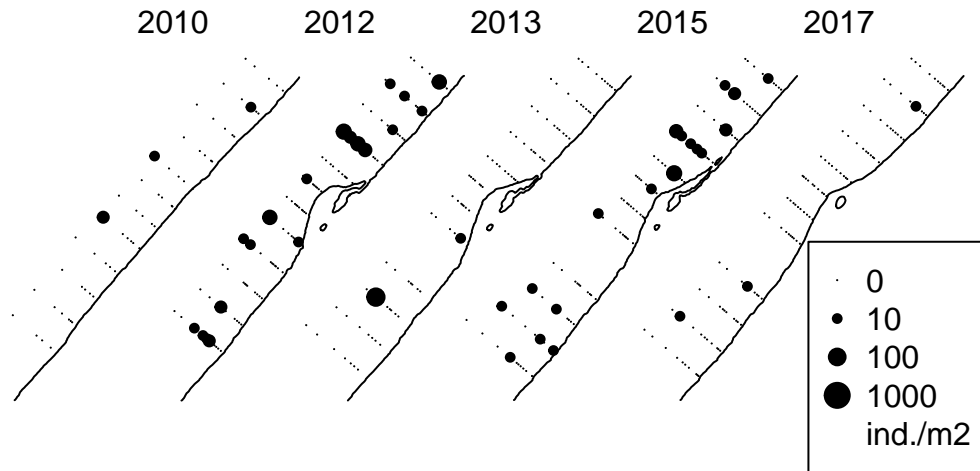
Scolecopsis bonnieri



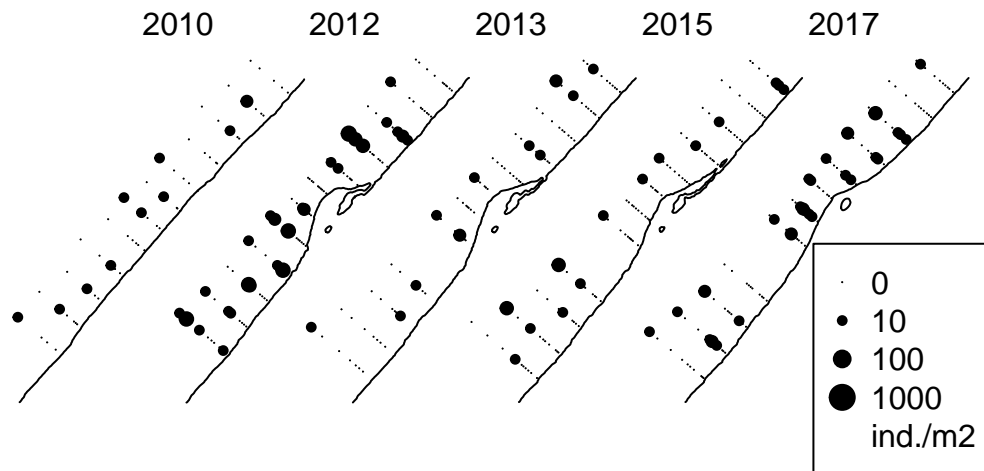
Echinocardium cordatum



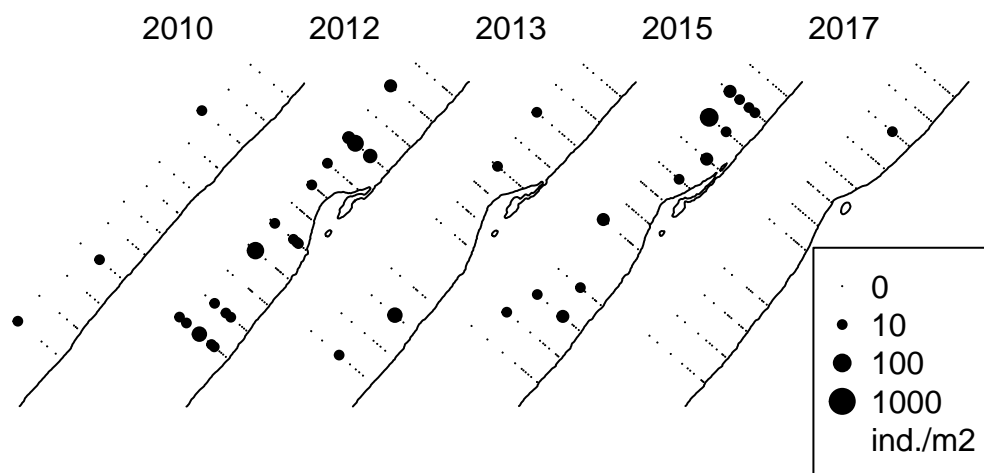
Nototropis swammerdamei



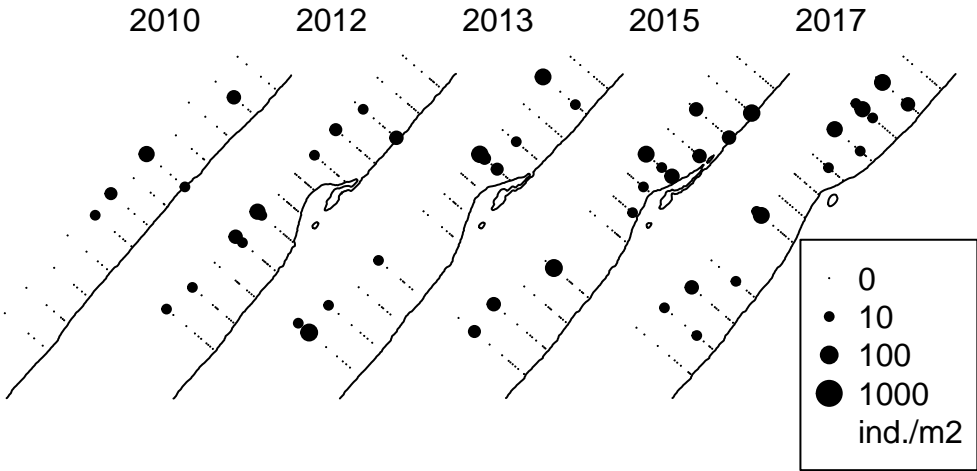
Crangon crangon



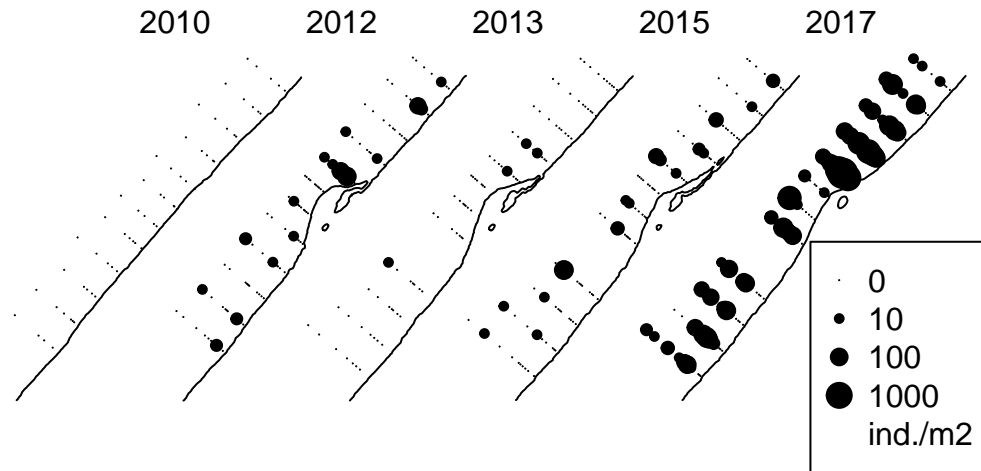
Phyllodoce (Anaitides) groenlandica



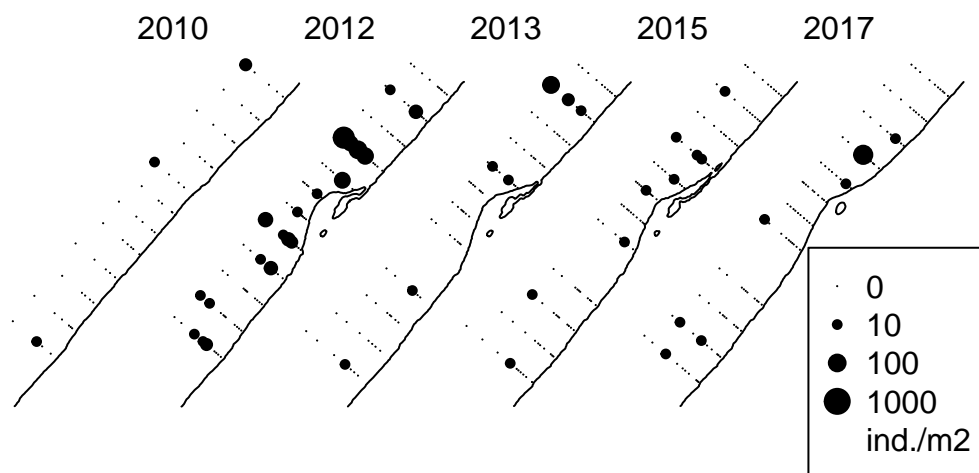
Urothoe poseidonis



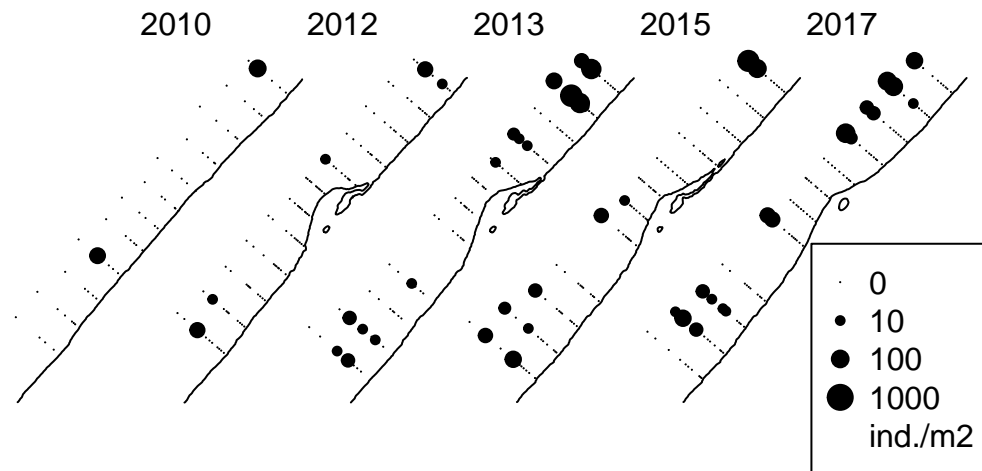
Donax vittatus



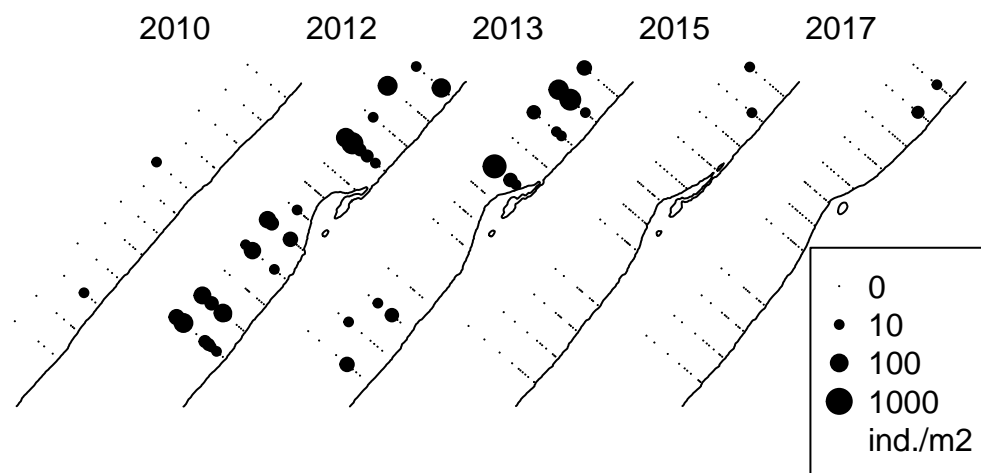
Ophiuroidea



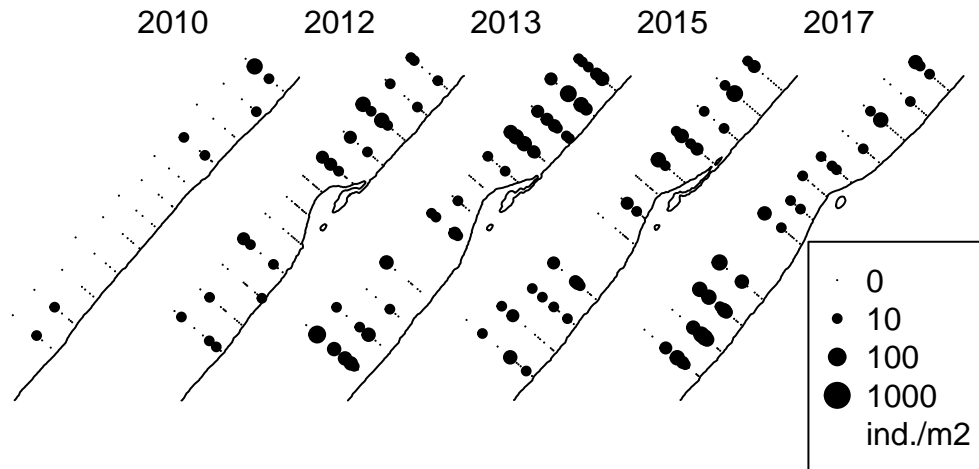
Oligochaeta



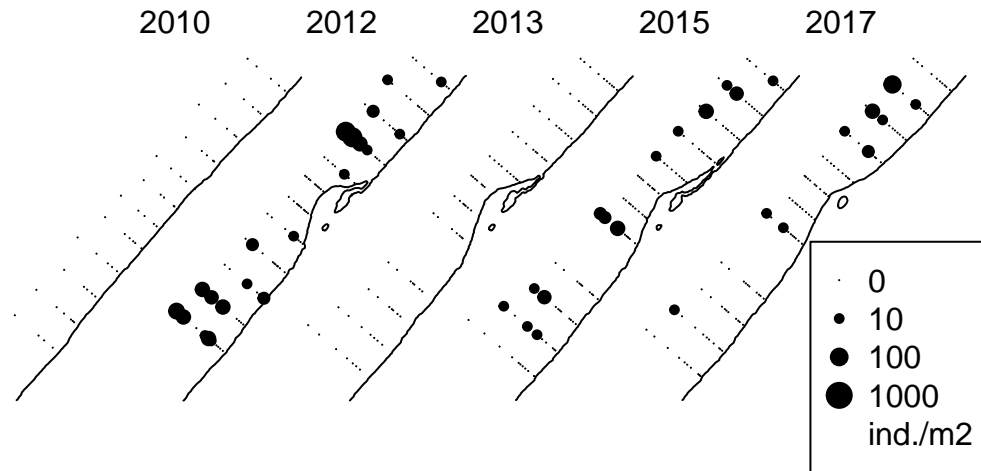
Malmgrenia



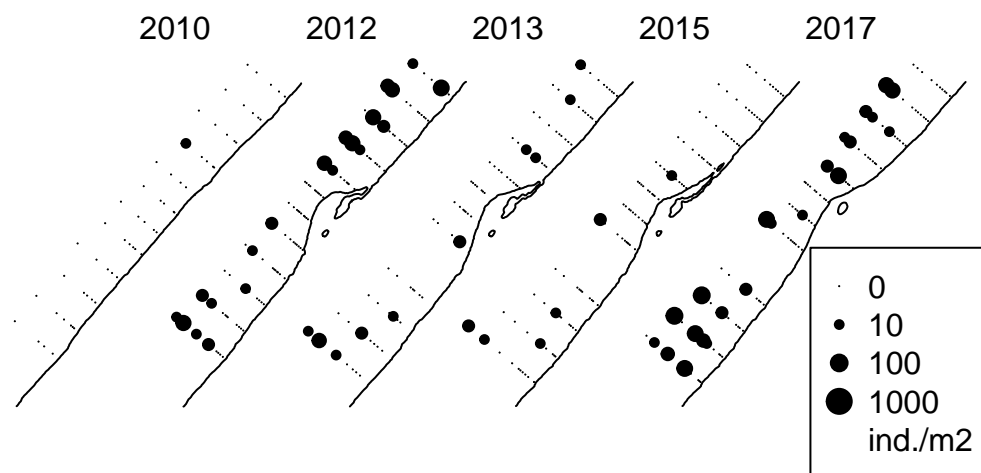
Diastylis bradyi



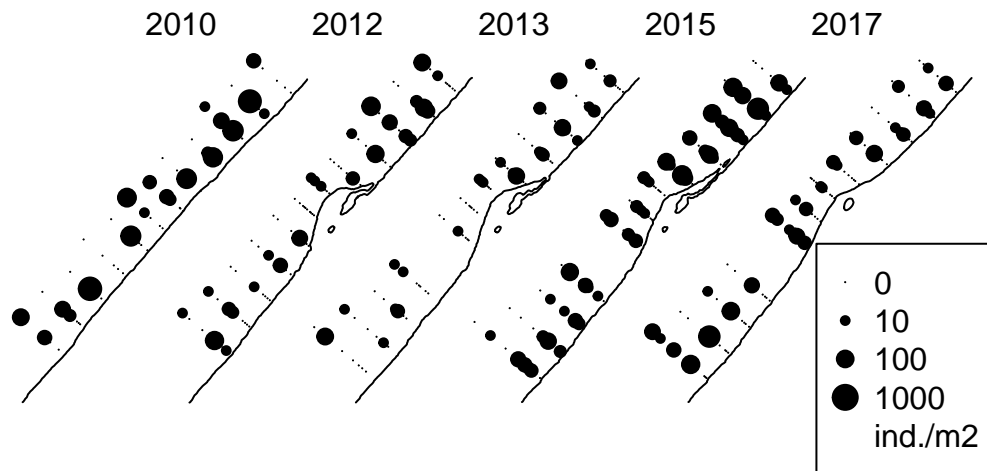
Poecilochaetus serpens



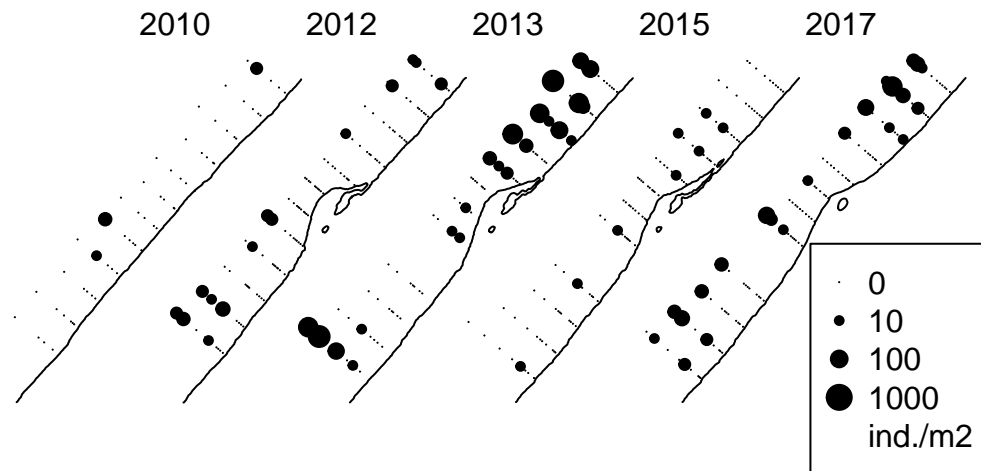
Leucothoe incisa



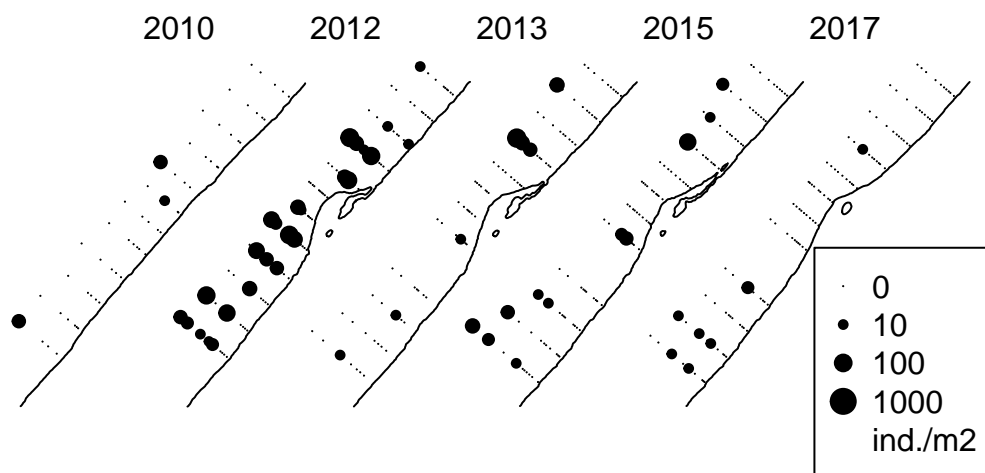
Magelona mirabilis



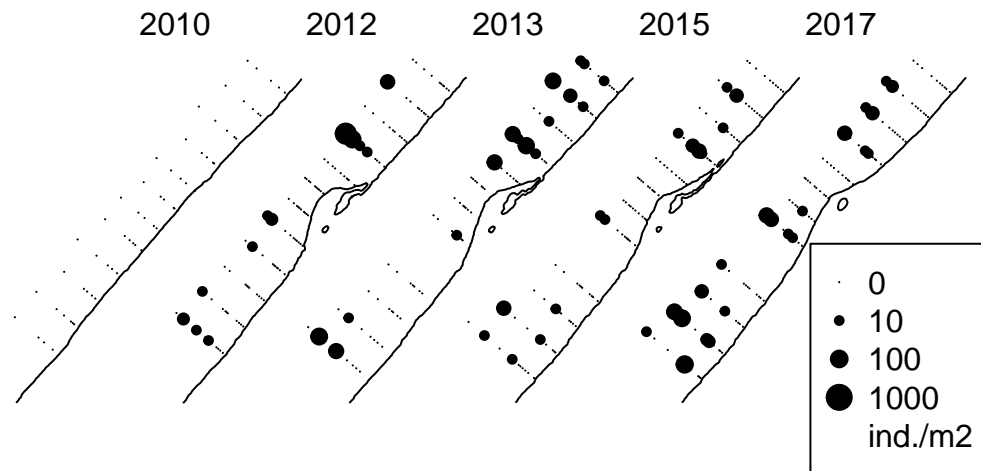
Malmgrenia darbouxi



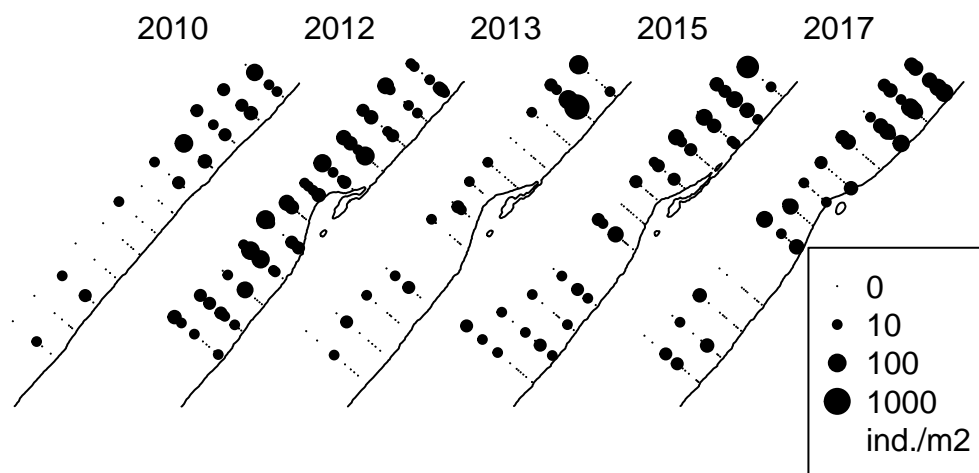
Lagis koreni



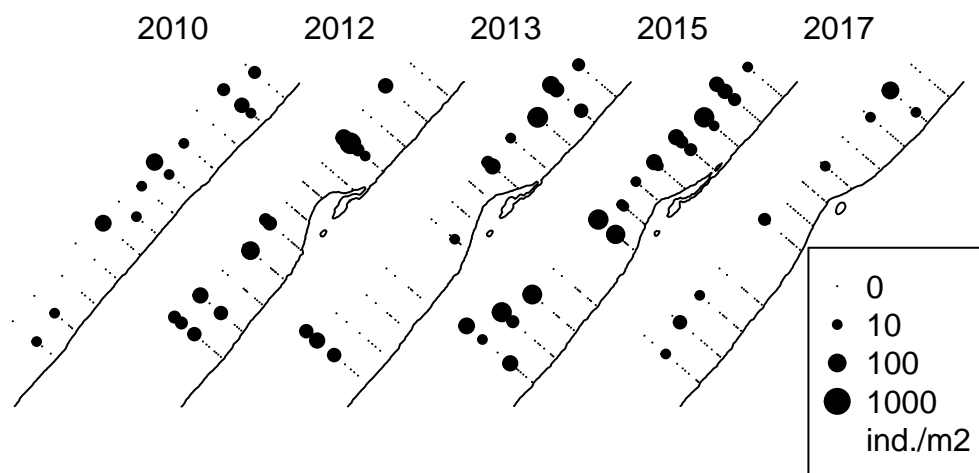
Caprellidae



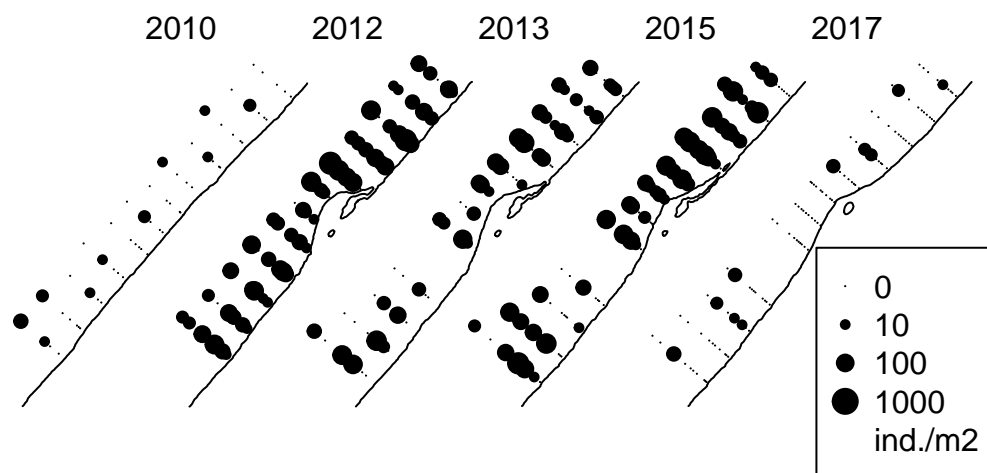
Nemertea



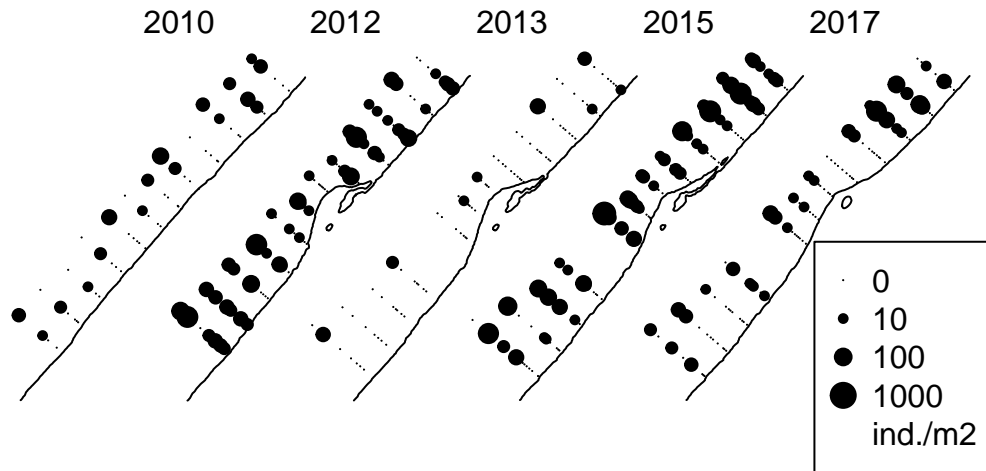
Notomastus latericeus



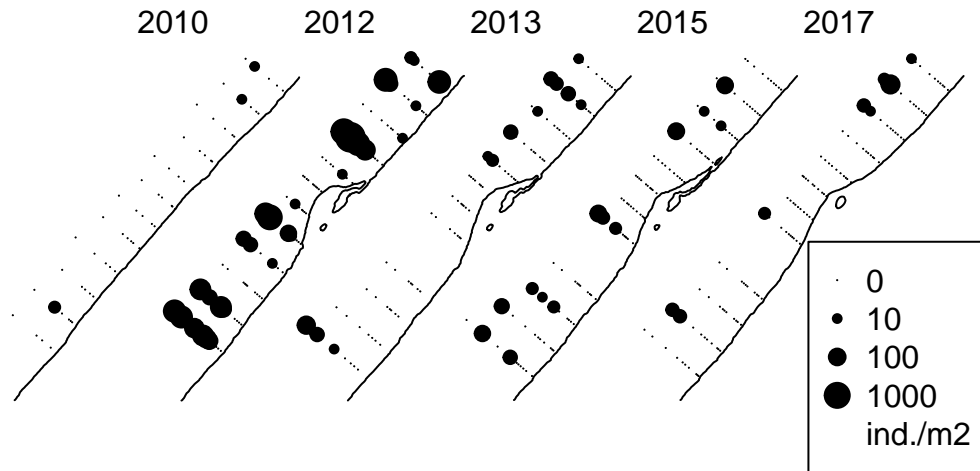
Nephtys



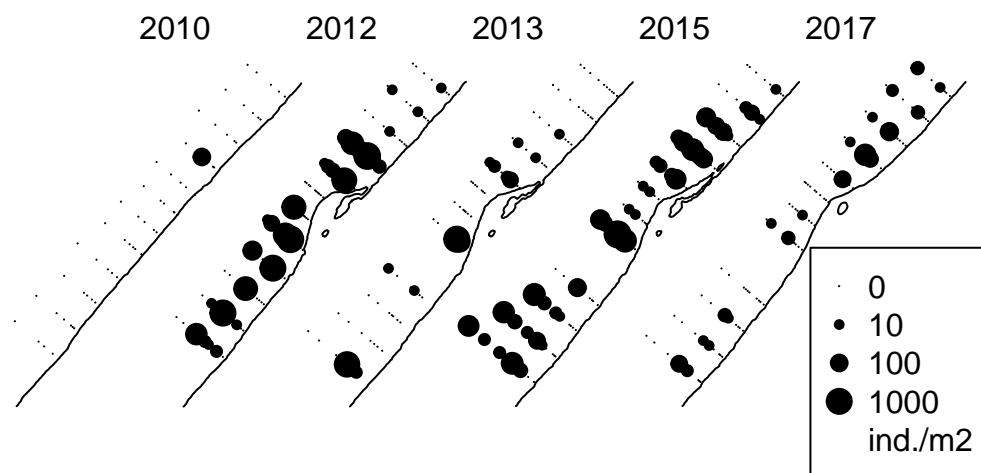
Eteone



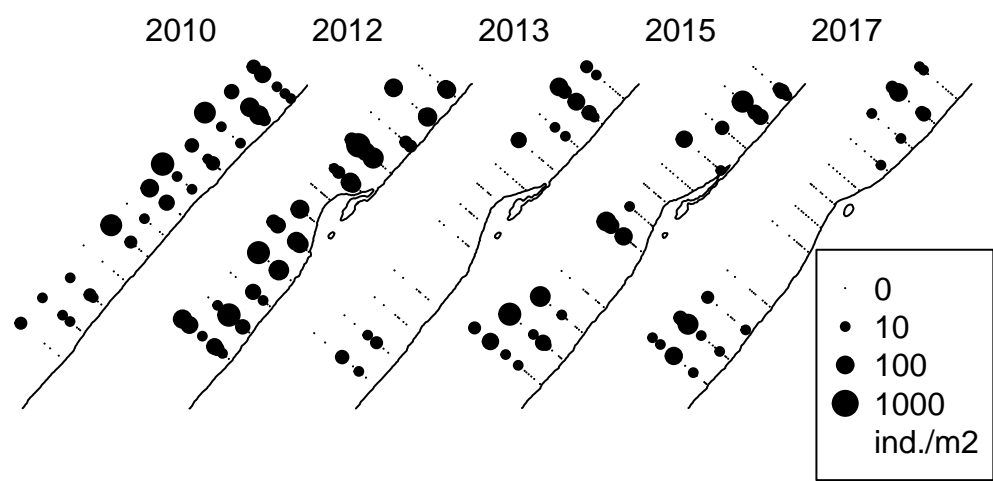
Eumida



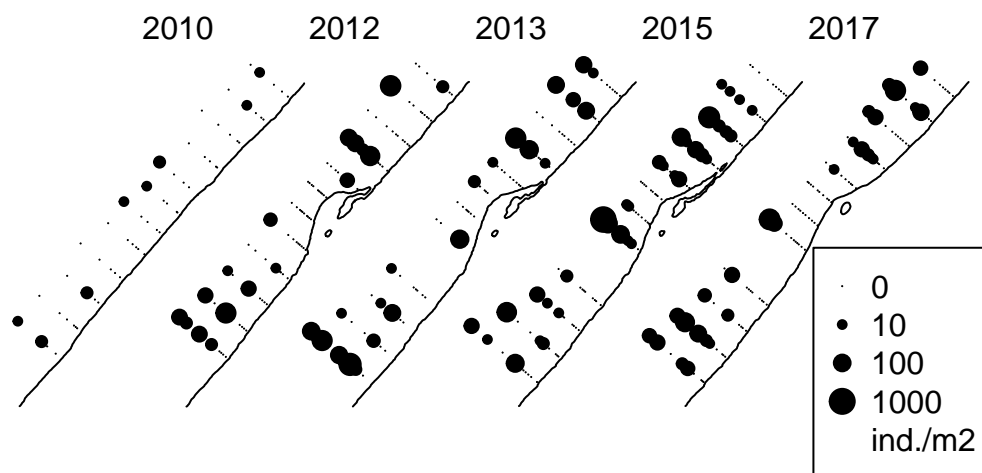
Macoma balthica



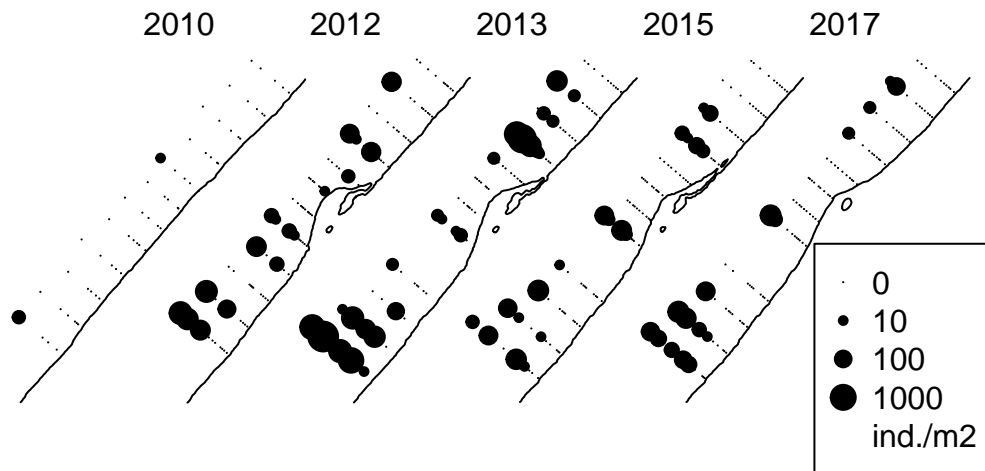
Phyllodoce mucosa



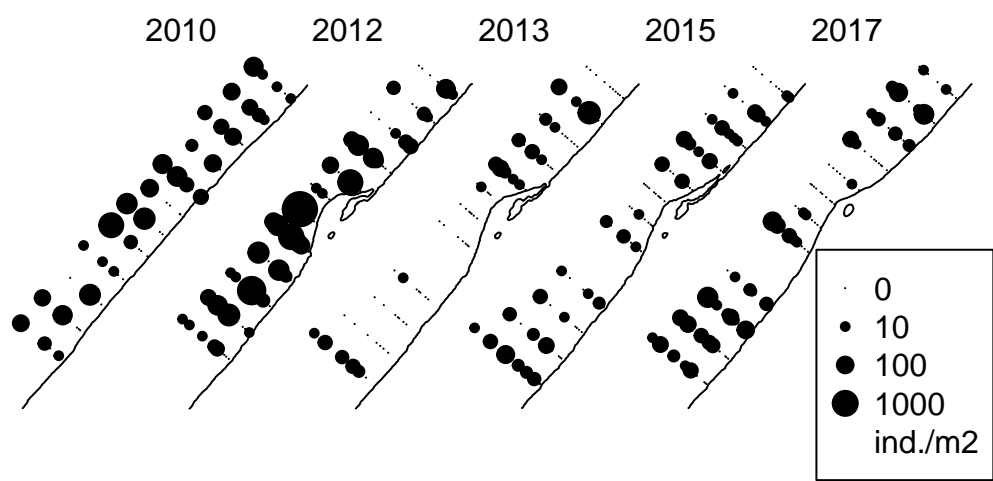
Kurtiella bidentata



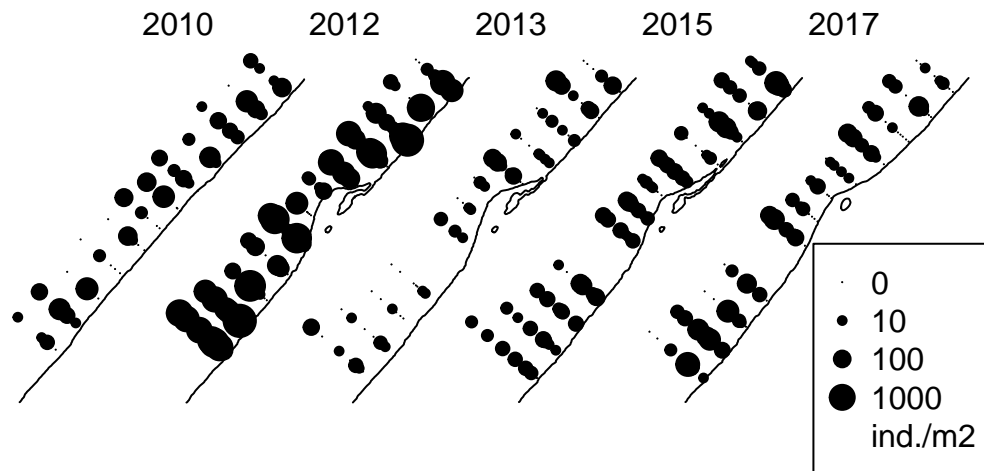
Owenia



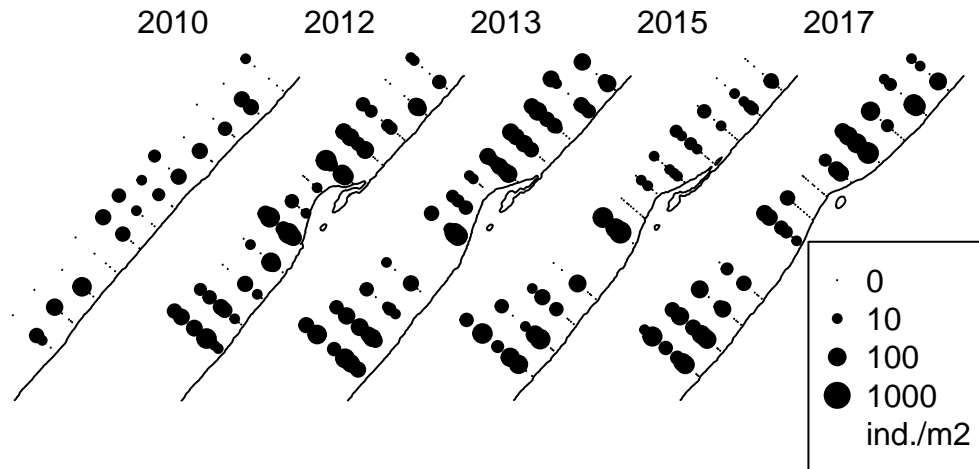
Capitella



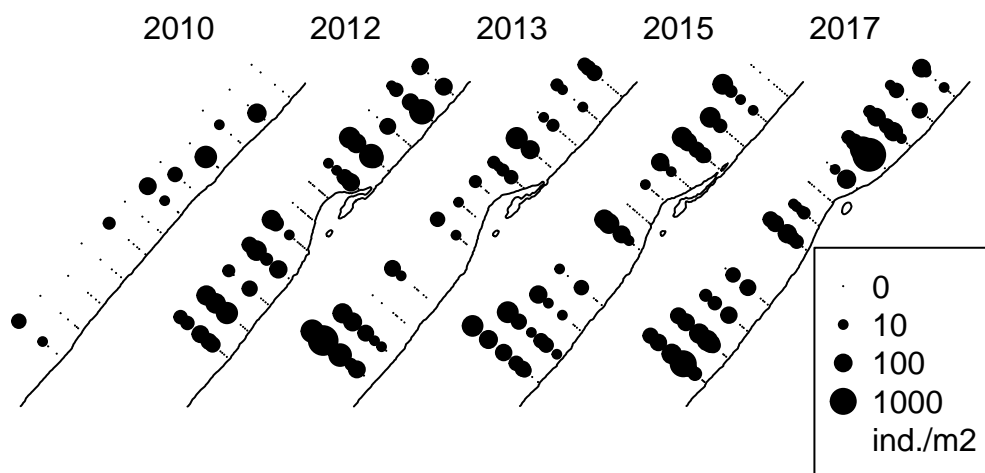
Spio



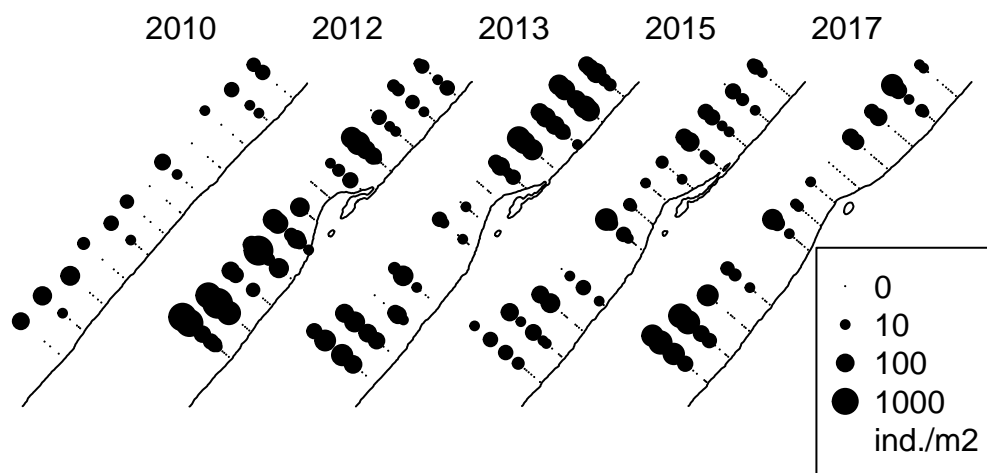
Nephtys hombergii



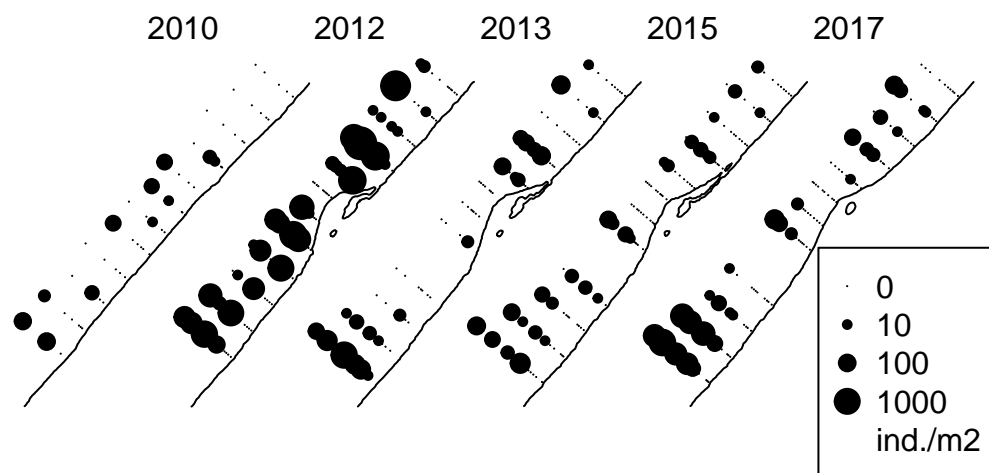
Spisula subtruncata



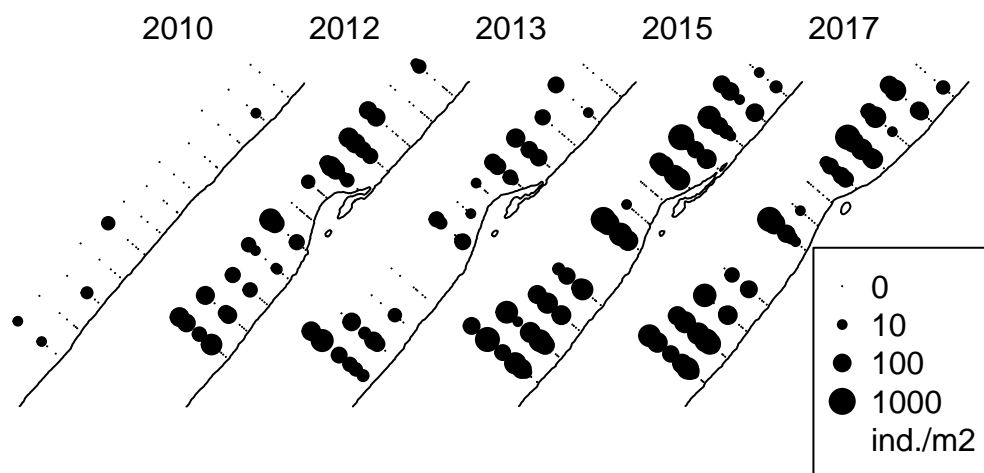
Scoloplos armiger



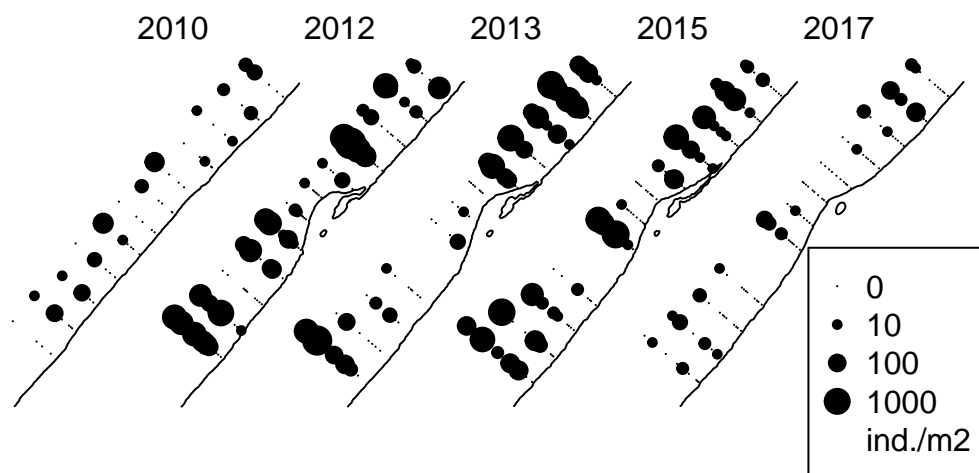
Abra alba



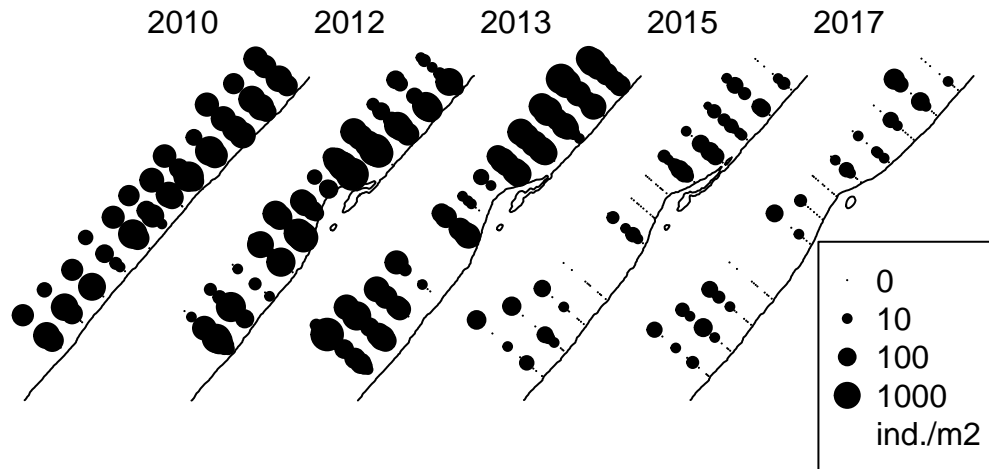
Fabulina fabula



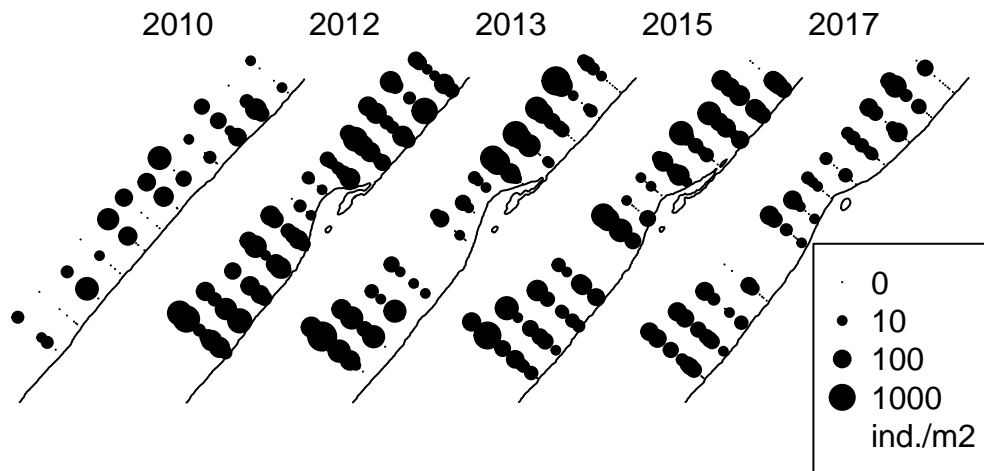
Lanice conchilega



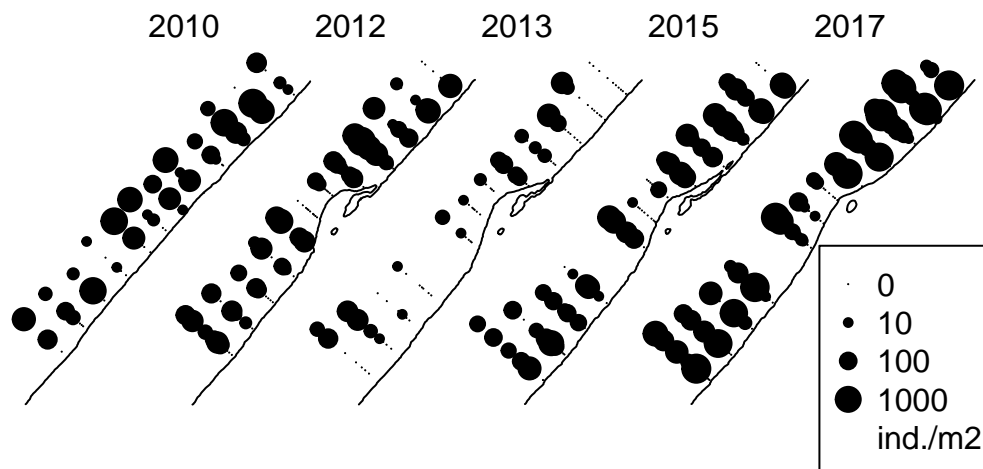
Ensis



Spiophanes bombyx



Magelona johnstoni



```
# we test the probability that the species have the same chance of being detected
# by all labs
# A complication is that none of the labs have worked in all years, and some species
# have trends over time
# We calculate the expected number of times a lab has observed a species as follows
#
# define labyear as the [nlab*nyear] matrix of fraction of total samples in that year j
# sorted by lab i
# define yearspec as the [nyear*nspec] matrix of frequency of species per year
#
# then the expected number of observations of species j by lab i is given as the
# matrix multiplication of labyear %*% yearspec
#
# This expected frequency is compared in a chi-square test with the observed number of
# observations of a species by a specific lab (in matrix labfreq)
#
# we only print the full chi-square test if this is (approximately) significant
#
smpca$lab[smpca$lab=="Eurofins"]<-"GRONTMIJ"
smpca$lab[smpca$lab=="IMARES_Ye"]<-"WMR"
smpca$lab[smpca$lab=="IMARES"]<-"WMR"
smpca$lab[smpca$lab=="IMARES_DH"]<-"WMR"
#
ul<-unique(smpca$lab)
nlab<-length(ul)
njaar<-5
```

```

nspec<-length(fr_sp_pca$freq)
jaars<-c(2010,2012,2013,2015,2017)
labyear<-matrix(nrow=nlab,ncol=njaar)
yearsvec<-matrix(nrow=njaar,ncol=nspec)
labfreq<-matrix(nrow=nlab,ncol=nspec)

# matrix labyear: a matrix with labs=rows, years=columns, expressing the fraction of
# samples of the year j treated by lab i
for(j in 1:njaar){
  for(i in 1:nlab){
    labyear[i,j]<-length(smpca$lab[smpca$lab==ul[i]&smpca$jaar==jaars[j]])
  }
}
jrsums<-apply(labyear,2,sum)
for(j in 1:njaar)labyear[,j]<-labyear[,j]/jrsums[j]

# matrix yearsvec, expressing per year the total frequency of each species
for(i in 1:nrow(fr_sp_pca)){
  nam<-row.names(fr_sp_pca)[i]
  for (j in 1:njaar){
    x<-dens[smpca$jaar==jaars[j],nam]
    yearsvec[j,i]<-length(x[x>0])
  }
}

# matrix labfreq, expressing per lab and species the number of times this lab has found
# this species
for(i in 1:nrow(fr_sp_pca)){
  nam<-row.names(fr_sp_pca)[i]
  for(j in 1:nlab){
    labnam<-ul[j]
    labfreq[j,i]<-length(smpca$lab[smpca$lab==labnam & dens[,nam]>0])
  }
}

# expected frequency of the species per lab, matrix multiplication of labyear and yearsvec
expfreq<-labyear %*% yearsvec

# do the check for all species
for(i in 1:nrow(fr_sp_pca)){
  nam<-row.names(fr_sp_pca)[i]
  pvec<-expfreq[,i]/sum(expfreq[,i])
  obsvec<-labfreq[,i]
  ct<-chisq.test(obsvec,p=pvec)
  if(is.na(ct$p.value))ct$p.value<- 1e-10
  if(ct$p.value>0.1){
    print(paste(nam," - No significant difference - p > 0.10"))
  }else{
    print(paste(nam))
    print(paste("Total no. obs",fr_sp_pca$freq[i]))
    print("Observed")
    print(paste(ul,obsvec))
    print("Expected")
  }
}

```

```
    print(paste(ul,round(pvec*fr_sp_pca$freq[i],0)))
    print(ct)
  }
}
par(mfrow=c(1,1))
```

Save smpca

```
# save smpca
save(smpca,file="./output/binary_files/smpca.Rdata")
write.csv(smpca,file="./output/csv_output/spmca.csv",row.names = F)
```