Ecostar WP4 - Data Preparation

Clement Garcia

2024-03-15

Formatting biological data

The original source of the biological data are stored in data_raw, from which the Essex team (UoE) did a thorough clean, standardisation and various formatting, the details of which are fully explained in the "data_processed/README.md".

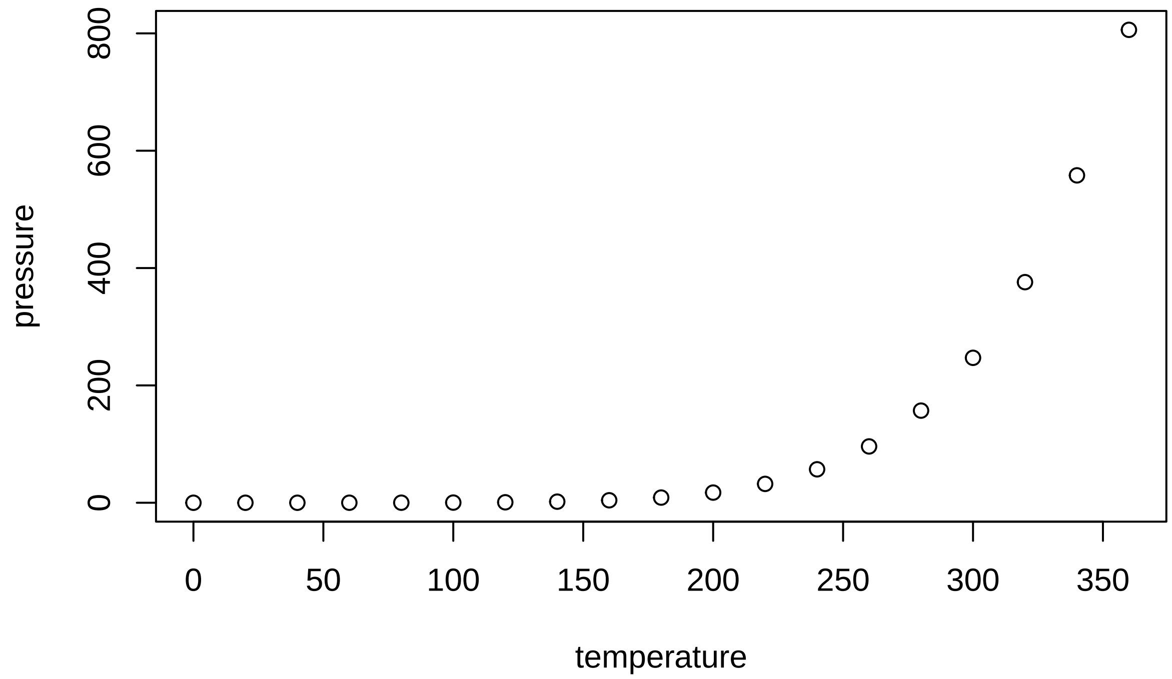
```r
## Adding Juvenile data * not to be done again
nameCor<-read.csv("./Name_Correction.csv")
SITE<-UKB_speMaster %>%
  select(NO,SITE) %>%
  distinct(SITE, .keep_all=T)

check<-left_join(SITE, nameCor)
check2<-check[grep("(juv)", check$SPECIES), c("SITE", "SPECIES")]
#write.csv(check2, "./Essex/Site_Check.csv")

siteChck<-read.csv("./Site_Check.csv")
siteChckName<-read.csv("./SiteCheckName.csv")
siteChckName$lifeCor<-"juvenile"
temps<-left_join(siteChck, siteChckName)
temps<-temps %>%
  distinct(SITE, NEW_SPECIES,  lifeCor)


## Formatting
speM<-UKB_speMaster[, c("new_species", "habitat", "habitat_class", "feeding_strat")]
speM<-left_join(speM, UKB_speList)
speM<-speM[, c("new_species", "acceptedName", "aphiaID", "kingdom", "phylum", "class", "order",
               "family",  "genus", "subgenus",  "species", "speciesDetails", "habitat", "habitat_class"

speSite<-UKB_Data[, c("SITE", "SPECIES", "abundance", "SURVEY", "YEAR", "MONTH", "STAT_NO", "SEASON")]
colnames(speSite)<-c("site", "new_species", "abn", "survey", "year", "month", "stat_no", "season")

spe<-left_join(speSite, speM)
spe<-spe %>%
  group_by(site, survey, year, month, stat_no, season, acceptedName, aphiaID, kingdom, phylum, class, o
           genus, subgenus,  species, speciesDetails, habitat, habitat_class, feeding_strat) %>%
  summarise(A = mean(abn, na.rm=T))


spe<-left_join(spe, temps, by = c("site" = "SITE", "acceptedName" = "NEW_SPECIES"))
```

```r
spePlat<-spe[!spe$lifeCor %in% "juvenile",]
spePlat<-spe[!is.na(spe$acceptedName),]
spePlat<-as.data.frame(spePlat)
spePlat$taxoLevel = with(spePlat,
                         ifelse(!is.na(species), "species",
                                ifelse(!is.na(genus), "genus",
                                       ifelse(!is.na(family), "family",
                                              ifelse(!is.na(order), "order",
                                                     ifelse(!is.na(class), "class",
                                                            ifelse(!is.na(phylum), "phylum",
                                                                   ifelse(!is.na(kingdom), "kingdom", NA


## Add average biomass
bMass<-read.csv("./body_mass_gWW_FINAL.csv")

##*Body Mass matching
#Unique taxa record from main database
taxa_unique<-spePlat %>%
  distinct(acceptedName, kingdom, phylum, class, order, family, genus, species, taxoLevel)
colnames(taxa_unique)[1]<-"ScientificName_accepted"
#Join the taxa with their respective recorded body mass
raw_gww<-left_join(taxa_unique, bMass[bMass$lvlEstimate %in% "taxa",], by = "ScientificName_accepted")
## Keep what has a direct match
spe_gww<-raw_gww[!is.na(raw_gww$Meanbodymass),]
spe_gww$taxoMassLvl<-"asRecorded"
## First leftovers
speNA<-raw_gww[is.na(raw_gww$Meanbodymass),]
## Species or Genus (to avoid the NA match)
gen_temp<-speNA[speNA$taxoLevel %in% c("species", "genus"),]
## Second left over (first part) what is not species or genus
genNA<-speNA[!speNA$taxoLevel %in% c("species", "genus"),]
## Join the taxa with match at Genus level
gen_gww<-left_join(gen_temp[, -which(names(gen_temp) %in% c("Meanbodymass","lvlEstimate"))],
                   bMass[bMass$lvlEstimate %in% "genus",],
                   by = c("genus" = "ScientificName_accepted"))
## Second left over (second part) no match added to the first part
genNA<-rbind(genNA, gen_gww[is.na(gen_gww$Meanbodymass),])
## Keep what has a Genus match
gen_gww<-gen_gww[!is.na(gen_gww$Meanbodymass),]
gen_gww$taxoMassLvl<-"asGenus"
## Species, Genus or Family (to avoid the NA match)
fam_temp<-genNA[genNA$taxoLevel %in% c("species", "genus", "family"),]
## Third left over (first part) what is not species or genus or family
famNA<-genNA[!genNA$taxoLevel %in% c("species", "genus", "family"),]
## Join the taxa with match at Family level
fam_gww<-left_join(fam_temp[, -which(names(fam_temp) %in% c("Meanbodymass","lvlEstimate"))],
                   bMass[bMass$lvlEstimate %in% "family",],
                   by = c("family" = "ScientificName_accepted"))
## Second left over (second part) no match added to the first part
famNA<-rbind(famNA, fam_gww[is.na(fam_gww$Meanbodymass),])
## Keep what has a Family match
fam_gww<-fam_gww[!is.na(fam_gww$Meanbodymass),]
```

```
fam_gww$taxoMassLvl<-"asFamily"
## Species, Genus, Family, Order (to avoid the NA match)
ord_temp<-famNA[famNA$taxoLevel %in% c("species", "genus", "family", "order") & !is.na(famNA$order),]
## Fourth left over (first part) what is not species or genus or family or Order
ordNA<-famNA[is.na(famNA$order),]
## Join the taxa with match at Order level
ord_gww<-left_join(ord_temp[, -which(names(ord_temp) %in% c("Meanbodymass","lvlEstimate"))],
                   bMass[bMass$lvlEstimate %in% "order",],
                   by = c("order" = "ScientificName_accepted"))
## Second left over (second part) no match added to the first part
ordNA<-rbind(ordNA, ord_gww[is.na(ord_gww$Meanbodymass),])
## Keep what has a Order match
ord_gww<-ord_gww[!is.na(ord_gww$Meanbodymass),]
ord_gww$taxoMassLvl<-"asOrder"
## Species, Genus, Family, Order, Class (to avoid the NA match)
cla_temp<-ordNA[ordNA$taxoLevel %in% c("species", "genus", "family", "order", "class") & !is.na(ordNA$cl
## Fifth left over (first part) what is not species or genus or family or Order or Class
claNA<-ordNA[is.na(ordNA$class),]
## Join the taxa with match at Class level
cla_gww<-left_join(cla_temp[, -which(names(cla_temp) %in% c("Meanbodymass","lvlEstimate"))],
                   bMass[bMass$lvlEstimate %in% "class",],
                   by = c("class" = "ScientificName_accepted"))
## Second left over (second part) no match added to the first part
claNA<-rbind(claNA, cla_gww[is.na(cla_gww$Meanbodymass),])
## Keep what has a Class match
cla_gww<-cla_gww[!is.na(cla_gww$Meanbodymass),]
cla_gww$taxoMassLvl<-"asClass"
claNA$taxoMassLvl<-"noInfo"
###
#Final compilation
final<-rbind(spe_gww, gen_gww, fam_gww, ord_gww, cla_gww, claNA)


## Binding average body mass to the original data
body_mass<-final[, c("ScientificName_accepted", "lvlEstimate", "taxoMassLvl", "Meanbodymass")]

spePlat<-left_join(spePlat, body_mass, by = c("acceptedName" = "ScientificName_accepted"))
#check<-unique(spePlat2[is.na(spePlat2$Meanbodymass), "acceptedName"])
#taxa_unique[taxa_unique$ScientificName_accepted %in% check,]
#missing Foraminifera, Platyhelminthes, Entoprocta, Porifera, Ciliophora, & high level taxo - OK

#save here
#save(spePlat, file="C:/Users/cg05/OneDrive - CEFAS/Science/Project - Commercial/INSITE/INSITE II/analy
```

Extracting environmental data from available raster

There was not much environmental information in the original dataset, therefore we used available data from raster (Mitchell et al. 2019) was used in conjonction with platform coordinates to extract the relevant data.



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.