

# Rock Pool National BioBlitz 2025

Rosemary Greensmith

2025-06-04

The purpose of this report is to show an infographic of rocky shore species presence data collected by citizen scientists during The Rock Pool Project 2025 iNaturalist Bioblitz. The infographic displays the number of non-terrestrial records that were identified to species level, grouped into native and non-native species, across taxonomic ranks (Kingdom, Phylum and Class) using bar plots. This report also shows the code used to create the infographic.

## Loading and preparing the data

The following R code loads the data listed below:

- iNaturalist bioblitz 2025 data - accessed via API
- World Register of Marine Species (WoRMS) taxonomic data - accessed via API
- List of UK non-native marine species - locally stored ("Data/UK marine NNS.csv")

```
# Define project ID and API parameters
project_slug <- "brpc-national-bioblitz-2025-practice"

# Download data using the rinat package
inat_data <- get_inat_obs_project(project_slug)

# Convert observed_on to date-time for comparison
inat_data$updated_at <- ymd_hms(inat_data$updated_at)
inat_data$time_observed_at <- ymd_hms(inat_data$time_observed_at)

last_update <- max(inat_data$updated_at)
cat("Last update:", as.character(last_update), "\n")
```

## Loading iNaturalist bioblitz 2025 data

## Last update: 2025-06-04 16:08:38.033

```
# Create empty columns to bind with the inat_data dataframe for the new taxonomic data
taxon.kingdom = rep(NA, times = length(inat_data[,1]))
taxon.phylum = rep(NA, times = length(inat_data[,1]))
taxon.class = rep(NA, times = length(inat_data[,1]))
taxon.order = rep(NA, times = length(inat_data[,1]))
taxon.family = rep(NA, times = length(inat_data[,1]))
marine = rep(NA, times = length(inat_data[,1]))
brackish = rep(NA, times = length(inat_data[,1]))
freshwater = rep(NA, times = length(inat_data[,1]))
terrestrial = rep(NA, times = length(inat_data[,1]))

# Combine the inat_data dataframe with the new columns
inat_data = cbind(
  inat_data, taxon.kingdom, taxon.phylum, taxon.class, taxon.order, taxon.family,
  marine, brackish, freshwater, terrestrial)

# Fill in new inat_data columns with taxonomic information from WoRMS
l = length(inat_data[,1])
for (a in 1:l) {
  if (
    is.na(inat_data$taxon.rank[a]) == FALSE && inat_data$taxon.rank[a] == "species"
  ){

    # Split the scientific name into two character objects: genus and species
    binomClassNm = inat_data$taxon.name[a]
    binomClassNmSplit = strsplit(binomClassNm, "[ ]")

    genus = binomClassNmSplit[[1]][1]
    species = binomClassNmSplit[[1]][2]

    # Paste the genus and species names into the WoRMS API key and download relevant data
    api = paste("https://www.marinespecies.org/rest/AphiaRecordsByName/",
      genus,
      "%20",
      species,
```

```

        "?like=true&marine_only=false&extant_only=true&offset=1",
        sep = "")
taxonInfo = GET(api)
taxonInfoContent = httr::content(taxonInfo, as = 'text')

# Populate the new columns with taxonomic data downloaded from WoRMS
if(object.size(taxonInfoContent)>112) {
  taxonInfoContentJSON = jsonlite::fromJSON(taxonInfoContent)

  inat_data$taxon.kingdom[a] = taxonInfoContentJSON$kingdom[1]
  inat_data$taxon.phylum[a] = taxonInfoContentJSON$phylum[1]
  inat_data$taxon.class[a] = taxonInfoContentJSON$class[1]
  inat_data$taxon.order[a] = taxonInfoContentJSON$order[1]
  inat_data$taxon.family[a] = taxonInfoContentJSON$family[1]

  if(is.na(taxonInfoContentJSON$isMarine[1])==FALSE) {
    inat_data$marine[a] = taxonInfoContentJSON$isMarine[1]
  }
  if(is.na(taxonInfoContentJSON$isBrackish[1])==FALSE) {
    inat_data$brackish[a] = taxonInfoContentJSON$isBrackish[1]
  }
  if(is.na(taxonInfoContentJSON$isFreshwater[1])==FALSE) {
    inat_data$freshwater[a] = taxonInfoContentJSON$isFreshwater[1]
  }
  if(is.na(taxonInfoContentJSON$isTerrestrial[1])==FALSE) {
    inat_data$terrestrial[a] = taxonInfoContentJSON$isTerrestrial[1]
  }
} else {
  inat_data$taxon.kingdom[a] = "taxon info not retrieved"
}
}
}

```

Loading and integrating WoRMS taxonomic data into the inaturalist dataframe

```

# Load the non-native species list
non_native_species <- read.csv("Data/UK marine NNS.csv")

# Match observations against non-native species list

```

```
natbioblitz_nns <- subset(inat_data, taxon.id %in% non_native_species$inat_id)
cat("Number of non-native species records found:", nrow(natbioblitz_nns), "\n")
```

## Linking to Non-native Species List

```
## Number of non-native species records found: 74
```

## Infographic

The following R code creates bar plots to show the number of non-terrestrial records that were identified to species level across three taxonomic ranks (Kingdom, Phylum and Class), split into native and non-native groups.

**Stacked bar plot of records** The infographic styling is based on the design of The Rock Pool Project website. The colours used in the plots were defined using the hex codes of colours from The Rock Pool Project website (obtained using a colour picker tool).

```
# Define plot title names
plotTitles=c("Class","Phylum","Kingdom")

# Start new plot device and define number of plotting regions
par(mfrow=c(1,3))

# Create the bar plots. The Loop index numbers are backwards for ease of defining inat
for (a in 3:1){
  # Filter out the non-terrestrial species data and species not identified to species
  inat_data_filtered = inat_data
  inat_data_filtered = filter(inat_data_filtered,
                             marine == 1 | brackish == 1 | freshwater ==1)

  # Define column reference number for the name of each taxa group for each of the thr
  colRefINat_data_filtered=length(inat_data_filtered)-(5+a)

  # Sort each of the names alphabetically to allow accurate matching when native and n
  taxonNames = sort(unique(
    inat_data_filtered[,colRefINat_data_filtered]))
```

```

# Create empty matrix for bar plot data
df = matrix(0,nrow=1,ncol=length(taxonNames))
colnames(df)=taxonNames
rownames(df)=c("nonNative")

# Define column reference number for the name of each taxa group
colRefNatbioblitz_nns = length(natbioblitz_nns)-(5+a)

# Create frequency tables for native and non-native species
nativeTable = table(inat_data_filtered[,colRefINat_data_filtered])
nonNativeTable = table(natbioblitz_nns[,colRefNatbioblitz_nns])

# Populate the empty matrix for bar plot data with frequency values of each non-nati
for (b in 1:ncol(df)){
  for (c in 1:length(dimnames(nonNativeTable)[[1]])){
    if (colnames(df)[b] == dimnames(nonNativeTable)[[1]][c]){
      df[b] = nonNativeTable[c]
    }
  }
}

# Re-define the matrix object as having the rows from the native species frequency t
df = rbind(nativeTable,df)

# Subtract the non-native species frequencies from the native species frequency row
for (d in 1:ncol(df)) {
  df[1,d] = df[1,d]-df[2,d]
}

# Setting margins for each plot individually
if (a==3) {
  par(mar=c(8,5.5,4.5,6)+0.1,xpd=TRUE) # (Kingdom)
} else if (a==2){
  par(mar=c(8,1.0,4.5,4)+0.1,xpd=TRUE)# (Phylum)
} else if (a==1){
  par(mar=c(8,3.0,4.5,1)+0.1,xpd=TRUE)# (Class)
}

# Plot the bar plot
barplot(df,
  col=c("#00a6fb","#F79824"),
  horiz = TRUE, cex.names = 1.1,las = 1,border = FALSE,
  space=0.03,
  font.axis=1,

```

```

        cex.axis = 1.1,
        cex.lab = 1.1,
        xlab="Number of records",
        col.lab =c("#191d2d"))

# Plot titles
mtext(paste(plotTitles[a],sep=""),
      side = 3, adj = 0, line = -1,cex = 1.15,col=c("#191d2d"),font = 1.2)

# Legend
if(a==2){
  legend("topright", inset = c(0.01, 1.15),
        fill = c("#00a6fb","#F79824"),
        legend=c("Native species",
                  "Non-native species"), horiz = TRUE,cex = 1.1,border = FALSE)
}
}

# Outer plot title
mtext("Number of non-terrestrial records identified to species level",
      side = 3, line = -2.5, outer = TRUE,col = c("#191d2d"),font = 2,cex = 1.4)

# Plot subtitle
last_update <- max(inat_data$updated_at)
mtext(paste("Last update:",last_update,sep = " "),side = 3, line = -3.7,
      outer = TRUE,col = c("#0e6bff"),
      cex = 0.8,font = 3)

# Add a two tone border
box("outer", col="#0e6bff",lwd=7)

```

# Number of non-terrestrial records identified to species

Last update: 2025-06-04 16:08:38.033

