Data exploration and cleaning

This notebook looks for common sources of error and flags those records manual revision by the assessors who capture data from each country.

The output is a file showing the records that need manual review or corrections, if any

## Get data

Load required libraries:

library(tidyr)  
library(dplyr)  
library(utile.tools)  
library(stringr)  
library(ggplot2)

Get Kobo raw output data:

kobo\_output<-read.csv(file="International\_Genetic\_Indicator\_testing\_V\_4.0\_-\_latest\_version\_-\_False\_-\_2023-11-02-08-23-26.csv", sep=";", header=TRUE) %>%  
  
## add taxon column  
mutate(taxon=(utile.tools::paste(genus, species, subspecies\_variety, na.rm=TRUE))) %>%  
 # remove white space at the end of the name  
 mutate(taxon=str\_trim(taxon, "right"))

Filter out records which were marked as “not\_approved” in the manual Kobo validation interface (this means country assessors determined the is something wrong with that particular record).

# check if any species is flagged as "validation\_status\_not\_approved"  
kobo\_output %>%  
 filter(X\_validation\_status=="validation\_status\_not\_approved") %>%  
 select(country\_assessment, name\_assessor, taxon)

## country\_assessment name\_assessor taxon  
## 1 south\_africa Paulette Bloomer Panthera leo  
## 2 south\_africa Rob Tolcher Oryx gazella  
## 3 mexico Romina Miranda Méndez Pseudoeurycea lineola  
## 4 mexico Ana Weiger Ambystoma altamirani  
## 5 mexico Romina Miranda Méndez Castilleja tolucensis  
## 6 france Fleur Visser Columba palumbus  
## 7 mexico Vania Jiménez Lobato Cucurbita argyrosperma sororia  
## 8 france Fleur Visser Vipera ursinii ursinii  
## 9 france Gaëlle Brahy Halichoerus grypus

# omit those records from data:  
kobo\_output<- kobo\_output %>%  
 filter(X\_validation\_status!="validation\_status\_not\_approved")

### Filter out any sort of tests

# select likely columns to say "test"  
cols= c("name\_assessor", "email\_assessor", "genus", "species", "subspecies\_variety",  
 "scientific\_authority", "common\_name", "GBIF\_taxonID", "NCBI\_taxonID", "time\_populations")  
  
# check for "test" or "template" on any of them  
kobo\_output %>%   
 filter(if\_any(all\_of(cols), ~ grepl("test", .)) |  
 if\_any(all\_of(cols), ~ grepl("Template", .))) %>%   
 select(country\_assessment, name\_assessor, genus, species)

## country\_assessment name\_assessor genus species  
## 1 united\_kingdom test test test  
## 2 united\_kingdom test test test  
## 3 united\_kingdom test test test  
## 4 united\_kingdom test test test

# filter them out of dataset  
kobo\_output<- kobo\_output %>%   
 filter(if\_any(all\_of(cols), ~ !grepl("test", .))) %>%   
 filter(genus!="Template")

## Check for common data capture errors

### Number of populations

In the form, -999 was used to mark taxa with unknown number of extant populations. This was used because answering the question was mandatory, so leaving it blank wasn’t possible. We have to change -999 to NA.

kobo\_output<-kobo\_output %>%  
 mutate(n\_extant\_populations= na\_if(n\_extant\_populations, -999))

We can now explore how many populations per species are still extant (still existing! NOT extinct!)?

summary(kobo\_output$n\_extant\_populations)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 1.00 3.00 63.28 8.00 47364.00 19

table(kobo\_output$n\_extant\_populations)

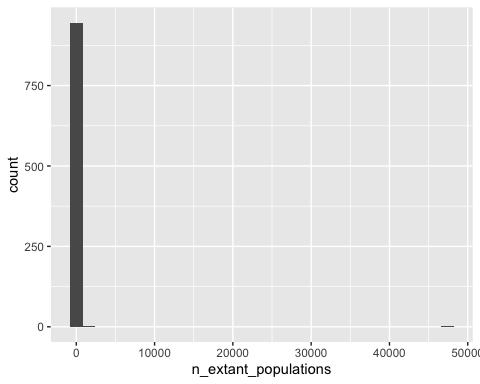
##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12   
## 4 303 134 86 76 41 36 15 23 14 21 13 9   
## 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 13 6 15 7 2 11 11 4 3 3 2 5 2   
## 26 27 28 30 33 34 35 36 37 38 40 42 43   
## 2 1 2 9 2 1 3 3 3 2 1 2 1   
## 45 47 48 50 53 54 55 57 60 65 68 69 72   
## 3 1 1 1 1 1 2 1 1 3 1 1 1   
## 74 75 77 79 80 85 87 89 91 95 100 104 105   
## 1 1 2 1 1 1 1 1 1 1 1 1 2   
## 113 116 122 124 127 137 146 148 150 173 176 180 186   
## 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 200 241 242 284 286 311 330 345 347 840 47364   
## 1 1 1 1 1 1 1 1 1 1 1

Plot histogram

ggplot(kobo\_output, aes(x=n\_extant\_populations))+  
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

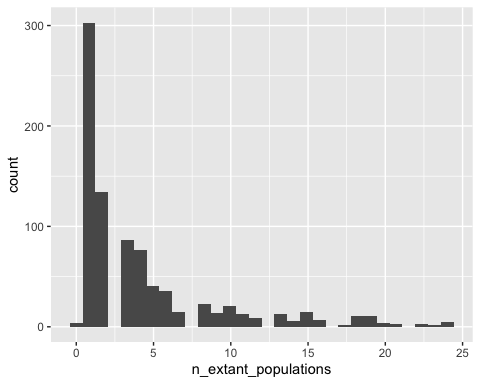
## Warning: Removed 19 rows containing non-finite values (`stat\_bin()`).



Zoom Plot histogram

kobo\_output %>%  
 filter(n\_extant\_populations>=0, n\_extant\_populations<25) %>%  
  
ggplot(., aes(x=n\_extant\_populations))+  
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Once -999 was replaced by NA there should be no negative number of populations (if they are, they are typos that need to be corrected).

kobo\_output %>%  
 filter(n\_extant\_populations<0) %>%  
 select(country\_assessment, taxon, name\_assessor, n\_extant\_populations, n\_extint\_populations)

## [1] country\_assessment taxon name\_assessor   
## [4] n\_extant\_populations n\_extint\_populations  
## <0 rows> (or 0-length row.names)

Show which taxa (if any) have 0 (zero) extant populations. **Is this correct? needs to be manually checked**

kobo\_output %>%  
 filter(n\_extant\_populations==0) %>%  
 select(country\_assessment, taxon, name\_assessor, n\_extant\_populations, n\_extint\_populations)

## country\_assessment taxon name\_assessor  
## 1 france Hieracium sandozianum Myriam Heuertz  
## 2 united\_states Planorbella magnifica Meg Mahoney  
## 3 sweden Charadrius alexandrinus Viktoria Köppä  
## 4 united\_states Heterelmis stephani Taylor Stack  
## n\_extant\_populations n\_extint\_populations  
## 1 0 1  
## 2 0 3  
## 3 0 3  
## 4 0 2

Show which species (if any) have 999 extant populations. **Should this be -999? OR n\_extinct pops??**

kobo\_output %>%  
 filter(n\_extant\_populations==999) %>%  
 select(country\_assessment, taxon, name\_assessor, n\_extant\_populations, n\_extint\_populations)

## [1] country\_assessment taxon name\_assessor   
## [4] n\_extant\_populations n\_extint\_populations  
## <0 rows> (or 0-length row.names)

Show which species (if any) have 999 EXTINCT populations. **Should this be -999?**

kobo\_output %>%  
 filter(n\_extint\_populations==999) %>%  
 select(country\_assessment, taxon, name\_assessor, n\_extant\_populations, n\_extint\_populations)

## [1] country\_assessment taxon name\_assessor   
## [4] n\_extant\_populations n\_extint\_populations  
## <0 rows> (or 0-length row.names)

Put all taxa with weird number of populations that need to be checked together:

check\_n\_pops <- kobo\_output %>%   
 # variables of interest  
 select(country\_assessment, name\_assessor, taxon, n\_extant\_populations, n\_extint\_populations) %>%  
  
 # same filters that discussed above  
 filter(n\_extant\_populations<0 |  
 n\_extant\_populations==0 |   
 n\_extant\_populations==999 |   
 n\_extint\_populations==999) %>%   
  
# add a column stating what needs to be checked:  
  
 mutate(need\_to\_check="check number of extant or extint populations. Are 0 correct? should 999 be -999? are extant/extint confused?")

### GBIF ID codes

Check GBIF

# check IDs  
head(kobo\_output$GBIF\_taxonID)

## [1] "2440940" "2337607" "2477927" "" "2422522" "5219516"

GBIF IDs tend to be 7 characters long. Some can be larger or shorter, but these seem to be exceptions. Therefore let’s flag any records where the GBIF Id is =/= 7 to manually check if it is correct.

kobo\_output %>%  
 filter(nchar(GBIF\_taxonID)>0, nchar(GBIF\_taxonID)!=7) %>%  
 # show only relevant columns  
 select(country\_assessment, name\_assessor, taxon, GBIF\_taxonID)

## country\_assessment name\_assessor taxon  
## 1 france Iris Lang Miniopterus schreibersii  
## 2 france Ivan Paz-Vinas Zingel asper  
## 3 australia Glenn Shea Phyllurus kabikabi  
## 4 australia Robyn Shaw Gymnobelideus leadbeateri  
## 5 australia Marlien van der Merwe Caladenia woolcockiorum  
## 6 mexico Eugenia Zarza Phonotimpus talquian  
## 7 australia Tanya Latty Leucopatus anophthalmus  
## 8 south\_africa Samantha Mynhardt Chersobius signatus  
## 9 australia Catherine Grueber Hypotaenidia sylvestris  
## 10 australia Rebecca Jordan Ambuchanania leuchbryoides  
## 11 united\_states Brenna Forester Anaxyrus williamsi  
## 12 australia Georgina Wood Ecklonia radiata brevipes  
## 13 colombia Victor Rincon-Parra Megascops gilesi  
## 14 colombia Victor Rincon-Parra Dubusia carrikeri  
## GBIF\_taxonID  
## 1 176678740  
## 2 172830100  
## 3 185106797  
## 4 220657  
## 5 185788620  
## 6 157318142  
## 7 10882478  
## 8 11030052  
## 9 10913878  
## 10 5792195; 8052578  
## 11 10897495  
## 12 177596818  
## 13 10784281  
## 14 10869060

Put them in their own happy df with a column stating what is the likely problem:

check\_GBIF <- kobo\_output %>%  
 filter(nchar(GBIF\_taxonID)>0, nchar(GBIF\_taxonID)!=7) %>%  
 # show only relevant columns  
 select(country\_assessment, name\_assessor, taxon, GBIF\_taxonID) %>%  
# add a column stating what needs to be checked:  
  
 mutate(need\_to\_check="check the GBIF taxonID. Either it looks plain different, or has more or less than 7 digits (most ids are 7 digits long, and this isn't, it could be an exception, or a mistake).")

### Species names

Genus, species and subspecies should be a single word, check if there are cases where it isn’t. Only exception would be “var.” or “subsp.” in the subspecies\_variety field:

kobo\_output %>%   
 filter(grepl(" ", genus) |   
 grepl(" ", species) |   
 grepl(" ", subspecies\_variety)) %>%  
 filter(!grepl("var.", subspecies\_variety)) %>%  
 filter(!grepl("subsp.", subspecies\_variety)) %>%  
 # show only relevant columns  
 # show only relevant columns  
 select(country\_assessment, name\_assessor, taxon, genus, species, subspecies\_variety)

## country\_assessment name\_assessor taxon  
## 1 sweden Viktoria Köppä Coregonus albula morphotype trybomi  
## genus species subspecies\_variety  
## 1 Coregonus albula morphotype trybomi

Put them in their own happy df with a column stating what is the likely problem:

check\_taxon\_names <- kobo\_output %>%   
 filter(grepl(" ", genus) |   
 grepl(" ", species) |   
 grepl(" ", subspecies\_variety)) %>%  
 filter(!grepl("var.", subspecies\_variety)) %>%  
 filter(!grepl("subsp.", subspecies\_variety)) %>%  
 # show only relevant columns  
 select(country\_assessment, name\_assessor, taxon, genus, species, subspecies\_variety) %>%  
 mutate(need\_to\_check="check genus, species or subspecies\_variety, we are targeting to have single words in each field, except in the ifraspecific names, where 'var.' and 'subsp.' (only) would be accepted. Other details or taxonomic notes should be added in the comments.")

## Create a single file for assessors review:

to\_check<-full\_join(check\_n\_pops, check\_GBIF) %>% full\_join(check\_taxon\_names) %>%  
 # show columns in desired order:  
select(country\_assessment, name\_assessor, taxon, need\_to\_check, n\_extant\_populations,  
 n\_extint\_populations, GBIF\_taxonID, genus, species, subspecies\_variety)

## Joining, by = c("country\_assessment", "name\_assessor", "taxon",  
## "need\_to\_check")  
## Joining, by = c("country\_assessment", "name\_assessor", "taxon",  
## "need\_to\_check")

# save file:  
  
write.csv(to\_check, "kobo\_output\_tocheck\_30april.csv", row.names = FALSE, fileEncoding = "UTF-8")

Session Info for reproducibility purposes:

sessionInfo()

## R version 4.2.1 (2022-06-23)  
## Platform: x86\_64-apple-darwin17.0 (64-bit)  
## Running under: macOS Big Sur ... 10.16  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] ggplot2\_3.4.1 stringr\_1.4.0 utile.tools\_0.2.7 dplyr\_1.0.9   
## [5] tidyr\_1.2.0   
##   
## loaded via a namespace (and not attached):  
## [1] highr\_0.9 pillar\_1.7.0 compiler\_4.2.1 tools\_4.2.1   
## [5] digest\_0.6.29 evaluate\_0.15 lifecycle\_1.0.3 tibble\_3.1.7   
## [9] gtable\_0.3.0 pkgconfig\_2.0.3 rlang\_1.0.6 cli\_3.6.0   
## [13] DBI\_1.1.3 rstudioapi\_0.13 yaml\_2.3.5 xfun\_0.31   
## [17] fastmap\_1.1.0 withr\_2.5.0 knitr\_1.39 generics\_0.1.3   
## [21] vctrs\_0.5.2 grid\_4.2.1 tidyselect\_1.1.2 glue\_1.6.2   
## [25] R6\_2.5.1 fansi\_1.0.3 rmarkdown\_2.14 farver\_2.1.1   
## [29] purrr\_0.3.4 magrittr\_2.0.3 scales\_1.2.0 ellipsis\_0.3.2   
## [33] htmltools\_0.5.5 assertthat\_0.2.1 colorspace\_2.0-3 labeling\_0.4.2   
## [37] utf8\_1.2.2 stringi\_1.7.6 munsell\_0.5.0 crayon\_1.5.1