

ProcessXLSXfromSEN.R

rstudio

Mon Mar 4 22:04:46 2019

```
##### Process xlsx from Stuart
# Routines to import and process xlsx files for ECC and Southrepps as supplied by SEN
# EAP 2019-02-14
# Files are called
#   2019-02-12_SN_EP_Classifier_results_SRepp.xlsx, and
#   2019-02-12_SN_EP_Classifier_results_Eccles.xlsx
## Update ## 2019-02-28 a further file recieved for Southrepps for 2016 - 2018,
## Update ## 2019-02-28, this file is called
#   2019-02-28_SN_EAP_Classifier_results_Southrepps2.xlsx
#
# Files are delivered as Microsoft Excel .xlsx format
# Files have the following structure
#   filename      : chr > character string, see below for details.
#   species       : chr > 6 character string with species code
#   confidence_index: num > Confidence index assigned by the classifier range 0.00 to 1.00 (0.99?)
#   real_error    : num > Relative Error calculated by the classifier range 0.00 to 1.00 (0.99?) See I
#
# Structure of values in the filename column:
#   XXX           : chr > 3 digit site code  SR2 = Southrepps (Dowlands), ECC == Eccls
#   _             : chr > separator
#   yyyyymmdd     : num > date of recording (assigned by SM2) ISO format
#   _             : chr > separator
#   hhmmss       : num > time of recording (assigned by SM2 no DST correction applied. hours since m
#   _             : chr > separator
#   NNN           : num > 3 digit number assigned by classifier, thought to be the call number in th
#
# Processing will be required to:
#   load data xlsx into a data.frame
#   extract date & time from filename
#   create new column fo observation date/time
#
# Input file are located in the following locations:
#   ~/R-Test/intermed/ECC, and
#   ~/R-Test/intermed/SR2
#
# Output files will be written to the following locations:
#   ~/R-Test/tidy/ECC, and
#   ~/R-Test/tidy/SR2
#
# Output files will have the following structure:
#   2019-02-12_SEN_Evaluation_XXX.csv : where XXX is either ECC or SR2 as relevant
#
##### CONFIG Follows #####

# Load required libraries

# Load tidyvers functions
```

```

#if (!require(tidyverse)) install.packages('tidyverse')
#library(tidyverse)
#library(readr)
#library(dplyr)
#library(purrr)
library(lubridate)
library(readxl) #Needed to process xls files
library(knitr)

#Evaluation parameters
re_threshold <-
  0.5 #Change this value to set required accuracy cut-off.
#In practice 0.5 is applied by Stuart when aggregating records.
save_csv <-
  TRUE #Change to FALSE if you don't want to create a new csv file
site_code <- "SR2" #See below for alternatives
input_file_name <-
  "2019-02-28_SN_EAP_Classifier_results_Southrepps2.xlsx"
output_file_name <- "2019-02-28_SEN_Evaluation_SR2.csv"

#Directories NB these are only valid for AWS - RStudio - Server
d_home <- "~/R-Test/"
d_raw <- paste(d_home, "raw/", site_code, "/", sep = "")
d_intermed <- paste(d_home, "intermed/", site_code, "/", sep = "")
d_tidy <- paste(d_home, "tidy/", site_code, "/", sep = "")
d_output <- paste(d_home, "output/", sep = "")

# Site Specific Information
validsitecodes <- c("SR2", "ECC")

##### CODE Follows #####
# Check if we have a valid site code
if (!(site_code %in% validsitecodes)) {
  stop("Invalid Site Code")
}

# Configure Environment & paths etc.
setwd(d_home)
getwd()

## [1] "/home/rstudio/R-Test"

# NB Following are hard coded paths rather than derived from site_code
tmp_sourcefile <-
  paste(d_intermed,
        input_file_name,
        sep = "")
tmp_outputfile <-
  paste(d_tidy,
        output_file_name,
        sep = "")

```

```

# Read the input file
tmp_SNclassifier_results <- read_excel(tmp_sourcefile)

tmp_SNclassifier_results <-
  data.frame(obs_datetime = as.POSIXct(gsub(
    "_", "", substr(tmp_SNclassifier_results$filename, 5, 19)
  ), format = "%Y%m%d%H%M%S"),
    tmp_SNclassifier_results) # Create the Date-Time column

#write results to csv
if (save_csv == TRUE) {
  write_csv(tmp_SNclassifier_results, tmp_outputfile, col_names = TRUE)
}

# Monthly Summary, results written to tbl_mnlyStats

tbl_mnlyStats <- tmp_SNclassifier_results %>%
  dplyr::filter(., real_error >= re_threshold) %>%
  group_by(year(as.Date(obs_datetime, "%Y-%m-%d")),
    month(as.Date(obs_datetime, "%Y-%m-%d")),
    species) %>%
  dplyr::summarise(
    count = n(),
    max = max(confidence_index),
    mean = round(mean(confidence_index), 2),
    min = min(confidence_index),
    std_dev = round(sd(confidence_index), 2)
  )
names(tbl_mnlyStats)[1] <- "Year"
names(tbl_mnlyStats)[2] <- "Month"
tbl_mnlyStats <- as.data.frame(tbl_mnlyStats)

#Now generate species summaries
species_found <- unique(tbl_mnlyStats$species)
print(paste(site_code, "Evaluation by SN"))

```

```

## [1] "SR2 Evaluation by SN"
for (row in 1:length(species_found)) {
  tmp_species <-
    filter(tbl_mnlyStats, species == species_found[row])
  print(knitr::kable(tmp_species))
}

```

```

##
##
## Year   Month species   count   max   mean   min   std_dev
## -----
## 2016     9  Barbar     15   0.99   0.93   0.48    0.15
## 2016    10  Barbar      6   0.99   0.99   0.99    0.00

```

```
## 2017      8 Barbar      3 0.99 0.99 0.99 0.00
## 2017      9 Barbar     13 0.99 0.97 0.84 0.04
## 2017     10 Barbar     24 0.99 0.95 0.70 0.07
## 2018      1 Barbar      1 0.99 0.99 0.99 NA
## 2018      3 Barbar      1 0.84 0.84 0.84 NA
## 2018      4 Barbar      1 0.17 0.17 0.17 NA
## 2018      7 Barbar      1 0.99 0.99 0.99 NA
## 2018      8 Barbar      7 0.99 0.88 0.52 0.18
## 2018      9 Barbar      7 0.99 0.99 0.98 0.00
## 2018     10 Barbar      7 0.99 0.99 0.99 0.00
## 2018     11 Barbar      1 0.98 0.98 0.98 NA
```

```
##
```

```
##
```

```
## Year   Month species   count   max   mean   min   std_dev
## -----
## 2016      9 Myodau      26 0.99 0.77 0.42 0.18
## 2016     10 Myodau      10 0.96 0.71 0.39 0.22
## 2016     11 Myodau       5 0.93 0.67 0.50 0.16
## 2017      3 Myodau       3 0.91 0.60 0.44 0.27
## 2017      8 Myodau      12 0.94 0.72 0.43 0.20
## 2017      9 Myodau      27 0.98 0.74 0.40 0.17
## 2017     10 Myodau      15 0.97 0.80 0.52 0.14
## 2017     11 Myodau       4 0.95 0.69 0.41 0.28
## 2018      1 Myodau       4 0.98 0.66 0.39 0.26
## 2018      3 Myodau       2 0.89 0.74 0.59 0.21
## 2018      4 Myodau       4 0.93 0.67 0.49 0.19
## 2018      5 Myodau       2 0.91 0.70 0.50 0.29
## 2018      7 Myodau       1 0.83 0.83 0.83 NA
## 2018      8 Myodau       9 0.98 0.85 0.63 0.14
## 2018      9 Myodau       5 0.95 0.87 0.73 0.10
## 2018     10 Myodau       6 0.93 0.76 0.46 0.17
```

```
##
```

```
##
```

```
## Year   Month species   count   max   mean   min   std_dev
## -----
## 2016      9 Myonat       4 0.99 0.85 0.78 0.09
## 2016     10 Myonat       5 0.98 0.88 0.76 0.10
## 2016     11 Myonat       2 0.99 0.94 0.88 0.08
## 2017      8 Myonat       3 0.99 0.97 0.94 0.03
## 2017      9 Myonat       7 0.99 0.95 0.85 0.05
## 2017     10 Myonat       7 0.99 0.90 0.80 0.07
## 2017     11 Myonat       7 0.99 0.92 0.79 0.08
## 2018      1 Myonat       4 0.98 0.81 0.72 0.12
## 2018      5 Myonat       2 0.96 0.84 0.71 0.18
## 2018      7 Myonat      37 0.99 0.99 0.88 0.02
## 2018      8 Myonat      48 0.99 0.97 0.72 0.05
## 2018      9 Myonat      19 0.99 0.97 0.89 0.03
## 2018     10 Myonat      22 0.99 0.96 0.79 0.05
## 2018     11 Myonat       5 0.96 0.86 0.72 0.13
```

```
##
```

```
##
```

```
## Year   Month species   count   max   mean   min   std_dev
## -----
## 2016      9 Nycnoc       7 0.99 0.92 0.72 0.09
```

##	2016	10	Nycnoc	2	0.76	0.70	0.65	0.08
##	2017	8	Nycnoc	18	0.99	0.85	0.63	0.13
##	2017	9	Nycnoc	20	0.99	0.83	0.64	0.12
##	2017	10	Nycnoc	19	0.99	0.93	0.68	0.09
##	2018	7	Nycnoc	64	0.99	0.92	0.63	0.09
##	2018	8	Nycnoc	44	0.99	0.90	0.65	0.10
##	2018	9	Nycnoc	12	0.99	0.84	0.63	0.13
##	2018	10	Nycnoc	18	0.99	0.91	0.69	0.08
##	2018	11	Nycnoc	3	0.99	0.91	0.77	0.12

##

##	Year	Month	species	count	max	mean	min	std_dev
##	-----	-----	-----	-----	-----	-----	-----	-----
##	2016	9	Pipnat	7	0.87	0.73	0.55	0.10
##	2017	9	Pipnat	2	0.82	0.78	0.74	0.06
##	2017	10	Pipnat	3	0.84	0.66	0.50	0.17
##	2018	4	Pipnat	2	0.85	0.70	0.56	0.21
##	2018	5	Pipnat	1	0.70	0.70	0.70	NA
##	2018	7	Pipnat	6	0.93	0.78	0.55	0.18
##	2018	8	Pipnat	4	0.97	0.85	0.74	0.10
##	2018	9	Pipnat	3	0.94	0.92	0.90	0.02
##	2018	10	Pipnat	3	0.87	0.65	0.52	0.19

##

##	Year	Month	species	count	max	mean	min	std_dev
##	-----	-----	-----	-----	-----	-----	-----	-----
##	2016	9	Pippip	106	0.99	0.93	0.38	0.12
##	2016	10	Pippip	43	0.99	0.94	0.51	0.10
##	2016	11	Pippip	11	0.98	0.91	0.39	0.17
##	2017	2	Pippip	1	0.97	0.97	0.97	NA
##	2017	3	Pippip	6	0.99	0.98	0.97	0.01
##	2017	4	Pippip	3	0.99	0.98	0.97	0.01
##	2017	5	Pippip	14	0.99	0.91	0.49	0.14
##	2017	6	Pippip	37	0.99	0.93	0.72	0.06
##	2017	7	Pippip	28	0.99	0.96	0.84	0.03
##	2017	8	Pippip	973	0.99	0.96	0.57	0.03
##	2017	9	Pippip	1567	0.99	0.96	0.41	0.04
##	2017	10	Pippip	187	0.99	0.95	0.55	0.08
##	2017	11	Pippip	26	0.99	0.96	0.83	0.04
##	2017	12	Pippip	2	0.98	0.98	0.98	0.00
##	2018	1	Pippip	22	0.99	0.97	0.93	0.02
##	2018	2	Pippip	4	0.98	0.96	0.93	0.02
##	2018	3	Pippip	11	0.99	0.98	0.96	0.01
##	2018	4	Pippip	29	0.99	0.94	0.62	0.08
##	2018	5	Pippip	103	0.99	0.95	0.61	0.06
##	2018	7	Pippip	665	0.99	0.92	0.38	0.12
##	2018	8	Pippip	262	0.99	0.92	0.45	0.10
##	2018	9	Pippip	69	0.99	0.95	0.57	0.08
##	2018	10	Pippip	97	0.99	0.95	0.60	0.07
##	2018	11	Pippip	27	0.99	0.95	0.81	0.05
##	2018	12	Pippip	4	0.99	0.98	0.95	0.02

##

##	Year	Month	species	count	max	mean	min	std_dev
----	------	-------	---------	-------	-----	------	-----	---------

```

## -----
## 2016      9  Pippyg      52  0.99  0.93  0.44  0.11
## 2016     10  Pippyg     11  0.99  0.93  0.71  0.09
## 2016     11  Pippyg      5  0.98  0.94  0.86  0.05
## 2017      4  Pippyg      1  0.88  0.88  0.88    NA
## 2017      5  Pippyg      3  0.98  0.93  0.84  0.08
## 2017      6  Pippyg      5  0.97  0.91  0.83  0.07
## 2017      7  Pippyg      2  0.97  0.96  0.94  0.02
## 2017      8  Pippyg     60  0.98  0.95  0.54  0.08
## 2017      9  Pippyg     78  0.99  0.95  0.57  0.06
## 2017     10  Pippyg     28  0.98  0.96  0.84  0.03
## 2017     11  Pippyg      6  0.98  0.95  0.91  0.03
## 2018      1  Pippyg      5  0.98  0.97  0.94  0.02
## 2018      3  Pippyg      2  0.98  0.98  0.98  0.00
## 2018      4  Pippyg      9  0.98  0.97  0.93  0.02
## 2018      5  Pippyg     19  0.99  0.91  0.37  0.15
## 2018      6  Pippyg      1  0.97  0.97  0.97    NA
## 2018      7  Pippyg    161  0.99  0.94  0.34  0.09
## 2018      8  Pippyg     77  0.99  0.88  0.33  0.16
## 2018      9  Pippyg     38  0.99  0.91  0.34  0.11
## 2018     10  Pippyg     46  0.98  0.93  0.57  0.09
## 2018     11  Pippyg      9  0.99  0.95  0.83  0.05
##
##
## Year  Month  species  count  max  mean  min  std_dev
## -----
## 2016      9  Pleaur     71  0.99  0.95  0.57  0.07
## 2016     10  Pleaur     22  0.99  0.96  0.88  0.03
## 2017      6  Pleaur      1  0.89  0.89  0.89    NA
## 2017      8  Pleaur     41  0.99  0.94  0.56  0.09
## 2017      9  Pleaur     48  0.99  0.94  0.69  0.08
## 2017     10  Pleaur     39  0.99  0.93  0.60  0.09
## 2018      1  Pleaur      1  0.99  0.99  0.99    NA
## 2018      2  Pleaur      2  0.94  0.92  0.91  0.02
## 2018      3  Pleaur     15  0.99  0.82  0.47  0.18
## 2018      4  Pleaur      6  0.99  0.88  0.39  0.24
## 2018      5  Pleaur      2  0.99  0.98  0.98  0.01
## 2018      7  Pleaur     50  0.99  0.96  0.73  0.06
## 2018      8  Pleaur     66  0.99  0.92  0.38  0.15
## 2018      9  Pleaur     25  0.99  0.90  0.41  0.15
## 2018     10  Pleaur     19  0.99  0.93  0.68  0.10
## 2018     11  Pleaur      2  0.99  0.97  0.95  0.03
##
##
## Year  Month  species  count  max  mean  min  std_dev
## -----
## 2017      8  Eptser      2  0.58  0.55  0.53  0.04
## 2017     10  Eptser      2  0.84  0.70  0.56  0.20
## 2018      7  Eptser     26  0.99  0.83  0.54  0.18
## 2018      8  Eptser     14  0.99  0.86  0.43  0.17
## 2018      9  Eptser      1  0.98  0.98  0.98    NA
##
##
## Year  Month  species  count  max  mean  min  std_dev

```

```

## -----
## 2017      8 Nyclei      1  0.79  0.79  0.79      NA
#### NOTE NOTE NOTE ####
#
# To generate a pdf report from this process it is not possible to use
# the RStudio ctr-K short cut as this throws a number of errors.
# Instead use the following code entered at the console
#
# rmarkdown::render(paste(d_home, "bin/snips/ProcessXLSXfromSEN.R", sep = ""), "pdf_document")
#
#### END END END ####

#### Supplementary code to combine the two SR2 files from SEN.
# Parameters
source_files <- c(
  "2019-02-12_SEN_Evaluation_SR2.csv",
  "2019-02-28_SEN_Evaluation_SR2.csv"
)

output_file_name <- "2019-03-04_Summary_SEN_Evaluation_SR2.csv"

tmp_file1 <- read_csv(paste(d_tidy, source_files[1], sep = ""))

## Parsed with column specification:
## cols(
##   obs_datetime = col_datetime(format = ""),
##   filename = col_character(),
##   species = col_character(),
##   confidence_index = col_double(),
##   real_error = col_double()
## )

tmp_file2 <- read_csv(paste(d_tidy, source_files[2], sep = ""))

## Parsed with column specification:
## cols(
##   obs_datetime = col_datetime(format = ""),
##   filename = col_character(),
##   species = col_character(),
##   confidence_index = col_double(),
##   real_error = col_double(),
##   actual_date = col_datetime(format = ""),
##   session_date = col_datetime(format = ""),
##   time = col_datetime(format = "")
## )

tmp_file1 <- tmp_file1 %>% transmute(
  .,
  obs_datetime = obs_datetime,
  filename = filename,
  species = species,
  confidence_index = confidence_index,
  real_error = real_error
)

```

```

tmp_file2 <- tmp_file2 %>% transmute(
  .,
  obs_datetime = obs_datetime,
  filename = filename,
  species = species,
  confidence_index = confidence_index,
  real_error = real_error
)

tmp_SNclassifier_results <- bind_rows(
  tmp_file1,
  tmp_file2,
  .id = NULL
)

write_csv(tmp_SNclassifier_results, paste(d_tidy, output_file_name, sep = ""), col_names = TRUE)

```