ProcessXLSXfromSEN.R

rstudio

Sun Feb 17 07:45:08 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
#
# 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
#
# Structure of values in the filename column:
#   XXX             : chr > 3 digit site code  SR2 = Southrepps (Dowlands), ECC == Eccls
#   _               : chr > separator
#   yyyymmdd        : 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 writen 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
library(readxl) #Needed to process xlxs files
# Load tidyvers functions
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
# Site Specific Information
validsitecodes <- c("SR2", "ECC")
site_code <- "SR2" #See below for alternatives


#Evaluation parameters
re_threshold <-
  0.5 #Change this value to set required accuracy cut-off.
      #In practice 0.5 is applied by Stuart when agreegating records.


#Directories  NB these are only vaid for AWS - RStudio - Server
d_home <-
  "/home/rstudio/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 = "")   #


######## CODE Follows ##########################
# Configure Environment & paths etc.
setwd(d_home)


# NB  Following are hard coded paths rather than derived from site_code
ECC_sourcefile <-
  paste(d_home,
        "intermed/ECC/2019-02-12_SN_EAP_Classifier_results_Eccles.xlsx",
        sep = "")
SR2_sourcefile <-
  paste(d_home,
```

```
        "intermed/SR2/2019-02-12_SN_EAP_Classifier_results_SRepp.xlsx",
        sep = "")
ECC_outputfile <-
  paste(d_home, "tidy/ECC/2019-02-12_SEN_Evaluation_ECC.csv", sep = "")
SR2_outputfile <-
  paste(d_home, "tidy/SR2/2019-02-12_SEN_Evaluation_SR2.csv", sep = "")


# Read the files
# NB These functions are site specific so have hard coded pathes
SN_classifier_results_SR2 <- read_excel(SR2_sourcefile)
SN_classifier_results_ECC <- read_excel(ECC_sourcefile)


# Create the Date-Time column
# NB These functions are site specific so have hard coded pathes
SN_classifier_results_SR2 <- data.frame(obs_datetime = as.POSIXct(gsub(
  "_", "", substr(SN_classifier_results_SR2$filename, 5, 19)
), format = "%Y%m%d%H%M%S"),
SN_classifier_results_SR2)
SN_classifier_results_ECC <- data.frame(obs_datetime = as.POSIXct(gsub(
  "_", "", substr(SN_classifier_results_ECC$filename, 5, 19)
), format = "%Y%m%d%H%M%S"),
SN_classifier_results_ECC)


#write results to csv
# NB These functions are site specific so have hard coded pathes
write_csv(SN_classifier_results_ECC, ECC_outputfile, col_names = TRUE)
write_csv(SN_classifier_results_SR2, SR2_outputfile, col_names = TRUE)


# Monthly Summary, results writen to tbl_mnlyStats
# There must be a better way to group by year/ month
if (!(site_code %in% validsitecodes))  {
  stop("Invalid Site Code")
} else {
  if (site_code == "ECC") {
    tmp_input <-  SN_classifier_results_ECC
  } else  if (site_code == "SR2") {
    tmp_input <-  SN_classifier_results_SR2
  }
  tbl_mnlyStats <- tmp_input %>%
    filter(., real_error >= re_threshold) %>%
    group_by(year(as.Date(obs_datetime, "%Y-%m-%d")),
             month(as.Date(obs_datetime, "%Y-%m-%d")),
             species) %>%
    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"))
  for (row in 1:length(species_found)) {
    tmp_species <-
      filter(tbl_mnlyStats, species == species_found[row])
    print(knitr::kable(tmp_species))

  }
}
```

```
## [1] "SR2 Evaluation by SN"
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Barbar         9   0.99   0.99   0.97      0.01
##  2015      11  Barbar         1   0.19   0.19   0.19        NA
##  2016       1  Barbar         1   0.19   0.19   0.19        NA
##  2016       2  Barbar         1   0.88   0.88   0.88        NA
##  2016       3  Barbar         4   0.44   0.34   0.26      0.08
##  2016       4  Barbar         2   0.44   0.34   0.24      0.14
##  2016       5  Barbar         2   0.99   0.99   0.99      0.00
##  2016       6  Barbar         3   0.98   0.75   0.30      0.39
##  2016       7  Barbar         2   0.99   0.76   0.53      0.33
##  2016       8  Barbar         1   0.99   0.99   0.99        NA
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Myodau         4   0.94   0.68   0.46      0.22
##  2015      11  Myodau         7   0.89   0.60   0.39      0.18
##  2015      12  Myodau         6   0.72   0.56   0.41      0.13
##  2016       1  Myodau         7   0.99   0.63   0.43      0.21
##  2016       2  Myodau        26   0.94   0.65   0.41      0.19
##  2016       3  Myodau        75   0.99   0.63   0.40      0.16
##  2016       4  Myodau        26   0.93   0.57   0.40      0.15
##  2016       5  Myodau        28   0.97   0.60   0.42      0.14
##  2016       6  Myodau         3   0.68   0.59   0.45      0.13
##  2016       7  Myodau        17   0.91   0.66   0.40      0.18
##  2016       8  Myodau         1   0.82   0.82   0.82        NA
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Myonat        12   0.99   0.90   0.73      0.09
##  2015      11  Myonat         5   0.99   0.94   0.86      0.06
##  2015      12  Myonat         7   0.99   0.88   0.72      0.11
```

```
##  2016       1  Myonat        11   0.96   0.84   0.72      0.08
##  2016       2  Myonat        27   0.99   0.90   0.71      0.09
##  2016       3  Myonat        18   0.99   0.86   0.73      0.10
##  2016       4  Myonat        10   0.99   0.92   0.82      0.07
##  2016       5  Myonat        24   0.99   0.91   0.70      0.10
##  2016       6  Myonat         4   0.98   0.95   0.92      0.03
##  2016       7  Myonat         5   0.98   0.91   0.82      0.07
##  2016       8  Myonat         3   0.99   0.93   0.87      0.06
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Nycnoc         2   0.97   0.96   0.96      0.01
##  2016       5  Nycnoc         1   0.74   0.74   0.74        NA
##  2016       7  Nycnoc        13   0.99   0.87   0.66      0.10
##  2016       8  Nycnoc         4   0.99   0.87   0.74      0.11
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Pipnat         2   0.75   0.70   0.64      0.08
##  2016       6  Pipnat         3   0.83   0.79   0.75      0.04
##  2016       7  Pipnat         1   0.87   0.87   0.87        NA
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Pippip       242   0.99   0.92   0.50      0.09
##  2015      11  Pippip        17   0.99   0.79   0.45      0.21
##  2015      12  Pippip         9   0.99   0.93   0.64      0.11
##  2016       1  Pippip        55   0.99   0.68   0.38      0.19
##  2016       2  Pippip        17   0.99   0.92   0.47      0.15
##  2016       3  Pippip        27   0.99   0.83   0.39      0.19
##  2016       4  Pippip        50   0.99   0.93   0.54      0.10
##  2016       5  Pippip       168   0.99   0.95   0.40      0.08
##  2016       6  Pippip       263   0.99   0.90   0.42      0.11
##  2016       7  Pippip       630   0.99   0.92   0.41      0.10
##  2016       8  Pippip       138   0.99   0.94   0.38      0.10
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Pippyg         2   0.98   0.97   0.96      0.01
##  2015      11  Pippyg         2   0.37   0.34   0.31      0.04
##  2016       1  Pippyg         6   0.97   0.74   0.30      0.27
##  2016       3  Pippyg         1   0.96   0.96   0.96        NA
##  2016       4  Pippyg         5   0.97   0.74   0.38      0.30
##  2016       5  Pippyg        46   0.99   0.94   0.71      0.06
##  2016       6  Pippyg         8   0.98   0.91   0.76      0.09
##  2016       7  Pippyg        25   0.98   0.88   0.49      0.15
##  2016       8  Pippyg        16   0.98   0.90   0.31      0.16
##
##
##  Year   Month  species    count    max   mean    min   std_dev
```

```
## -----  ------  --------  ------  -----  -----  -----  --------
##  2015      10  Pleaur        20   0.99   0.92   0.54      0.11
##  2016       2  Pleaur         2   0.81   0.58   0.35      0.33
##  2016       3  Pleaur         6   0.95   0.73   0.42      0.21
##  2016       4  Pleaur         4   0.98   0.82   0.44      0.26
##  2016       5  Pleaur        14   0.99   0.98   0.93      0.02
##  2016       6  Pleaur        71   0.99   0.93   0.44      0.11
##  2016       7  Pleaur        46   0.99   0.92   0.39      0.11
##  2016       8  Pleaur        27   0.99   0.93   0.61      0.10
##
##
##  Year   Month  species    count    max   mean    min   std_dev
## -----  ------  --------  ------  -----  -----  -----  --------
##  2016       8  Eptser         1   0.52   0.52   0.52        NA
```