bbsAssistant: An R package for downloading and handling data and information from the North American Breeding Bird Survey.

2019-07-11

Function Manual

For function descriptions and examples please see the manual.

Contributions

If you would like to contribute, please submit a pr or email (jessicaleighburnett at gmail). I am especially interested in having another set of eyes and hands to transfer and clean up the functions listed in Issue 1.

Overview of bbsAssistant Features

Installing package and loading dependencies

```r
# devtools::install_github("trashbirdecology/bbsAssistant", dependencies = TRUE, force=FALSE)
library(bbsAssistant)
library(rvest)
library(gdata)
library(feather)
library(here)
library(dplyr)
library(tidyverse)
```

Downloading the BBS data from USGS FTP

Define and/or create local directories

This function will create, if it does not already exist, folder **./bbsData/** within which to locally store BBS data and results. **NOTE**: If the directory exists, it will not overwrite files. If the bbs data already exists inside bbsDir, then we will create a logical to NOT download the data (see below). If you wish to download more, or overwrite existing data, please specify downloadBBSData=TRUE or remove .zip files from **/bbsData/**.

```r
# Create a directory to store and/or load the BBS data as feathers
bbsDir <- here::here("bbsData/")
dir.create(bbsDir)
```

Retrieve and import BBS data

If necessary, download all or some of the BBS state-level data. Note: Downloading all the data to file takes 10-15 minutes, so only run if you have not recently downloaded the BBS data.

Let's focus on a single species and state for brevity:

First, let's retrieve the regions of data that are available. The function `get_regions` retrieves the .zip filenames of all U.S. states and Canadian provinces, including their reference numbers and region codes.

```
# a. Load the regional .txt file from Patuxent's FTP server (you must be connected to the internet to p
regions <- get_regions()
```

Let’s restrict our data download to **Florida** data:

```
regionFileName <- regions$zipFileName %>% na.omit()
(regionFileName.use <- regionFileName[which(str_detect(regionFileName, "Flori")==TRUE)])
```

```
## [1] "Florida.zip"
```

Once we have one or more region filenames, we can use function `get_bbsData` to download the .zip file to a temporary folder (unless otherwise specified), and *import* the temp file to R object. The R object, flBBS, contains the raw BBS count data.

```
# check to see whether the data we need is already on file. if not, download it.
flBBS <- get_bbsData(file=regionFileName.use)
```

```
## [1] "Data were imported from the FTP server"
```

Next, we can download the BBS route-level geographic information and metadata, and append this to the original data.

```
routes <- get_routeInfo() # retrieve route-level data
flBBS <- left_join(flBBS, routes) # merge route-level data to bird count data
glimpse(flBBS %>% dplyr::select(aou, year, route, statenum, countrynum, stoptotal, latitude, longitude))
```

```
## Observations: 139,735
## Variables: 8
## $ aou        <int> 1840, 2000, 2010, 2890, 3100, 3131, 3160, 3390, 343...
## $ year       <int> 1967, 1967, 1967, 1967, 1967, 1967, 1967, 1967, 196...
## $ route      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ statenum   <int> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25,...
## $ countrynum <int> 840, 840, 840, 840, 840, 840, 840, 840, 840, 840, 8...
## $ stoptotal  <dbl> 3, 1, 1, 37, 1, 1, 7, 1, 1, 1, 1, 1, 2, 11, 5, 4, 2...
## $ latitude   <dbl> 30.92918, 30.92918, 30.92918, 30.92918, 30.92918, 3...
## $ longitude  <dbl> -87.40794, -87.40794, -87.40794, -87.40794, -87.407...
```

If we wish to save these data to file, we can do so by saving as **.feather*s, a compressed file formatted for use in R.

```
export_bbsFeathers(dataIn = flBBS,
                newDir   = bbsDir,
                filename = regionFileName.use)
```

Import BBS data from file into R

If the BBS data was downloaded previously and saved as .feather, we can import it using `import_bbsFeathers`. The code below is particularly useful if you are importing multiple files (e.g., multiple states)

```
(featherNames <- list.files(bbsDir, pattern = ".feather"))
```

```
## [1] "Florida.feather"
```

```
featherNames <- str_c("/", featherNames) #add separator

feather <- import_bbsFeathers(newDir   = bbsDir,
                              filename = featherNames)
glimpse(feather) # Notice that the data imported from disk (feathers) differs from the original BBS dat
```

```
## Observations: 139,735
## Variables: 9
## $ year       <int> 1967, 1967, 1967, 1967, 1967, 1967, 1967, 1967, 196...
## $ countrynum <int> 840, 840, 840, 840, 840, 840, 840, 840, 840, 840, 8...
## $ statenum   <int> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25,...
## $ route      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ bcr        <int> 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27,...
## $ latitude   <dbl> 30.92918, 30.92918, 30.92918, 30.92918, 30.92918, 3...
## $ longitude  <dbl> -87.40794, -87.40794, -87.40794, -87.40794, -87.407...
## $ aou        <int> 1840, 2000, 2010, 2890, 3100, 3131, 3160, 3390, 343...
## $ stoptotal  <dbl> 3, 1, 1, 37, 1, 1, 7, 1, 1, 1, 1, 1, 2, 11, 5, 4, 2...
```

Option for downloading ALL BBS data

If you wish to download and/or import ALL the data, you might choose to do so in a loop. Note: this is expensive! The following are not run in this example.

```
# Throw a warning if files exist
    if(length(list.files(bbsDir, pattern = "*.feather")) > 0 ){
        downloadBBSData = FALSE
    }else(
        {dir.create(bbsDir)
        downloadBBSData = TRUE}
        )
## Download ALL the regions of BBS data
if(downloadBBSData==TRUE){
for(i in 1:length(regionFileName)){
        bbsData <-  import_bbsData(
            # arguments for get_bbsData()
            file = regionFileName[i],
            dir =  "ftp://ftpext.usgs.gov/pub/er/md/laurel/BBS/DataFiles/States/",
            year = NULL, # subset by year
            aou = NULL, # subset by AOU #s
            countrynum = NULL, # subset by country number
            states = NULL, # subset by state/povince number
            #  arguments for get_routeInfo():
            routesFile = "routes.zip",
            routesDir =  "ftp://ftpext.usgs.gov/pub/er/md/laurel/BBS/DataFiles/",
            RouteTypeID = 1,
            # one or more of c(1,2,3)
            Stratum = NULL,  # subset by BBS stratum
            BCR = NULL # subset by BCR (bird conservation region)
        )
# d. Save the unzipped files to disk.
export_bbsFeathers(dataIn  = bbsData,
                newDir  = bbsDir,
                filename = regionFileName[i])
# e. Clear object from memory
rm(bbsData)
} # end section I. loop
}else(message(paste0("NOT DOWNLOADING BBS DATA. If you wish to download the BBS data, please remove fil
```

Use the same code as above to import *multiple* feathers from file:

```
feathers <- NULL
featherNames <- list.files(bbsDir, pattern = ".feather")
featherNames <- str_c("/", featherNames) #add separator
for (i in 1:length(featherNames)) {
  feather <- NULL
  feather <- import_bbsFeathers(newDir  = bbsDir,
                                filename = featherNames[i])
  feathers <- rbind(feathers, feather)
  rm(feather)

}
```

Subsetting the BBS count data

Subset BBS data by taxonomic groups

First, retrieve the species list from the BBS FTP server.

```
spp <- get_speciesList()
glimpse(spp)
```

```
## Observations: 750
## Variables: 9
## $ seq              <dbl> 6, 7, 8, 9, 10, 11, 13, 18, 19, 21, 22, 23, 2...
## $ aou              <dbl> 1770, 1780, 1760, 1690, 1691, 1700, 1710, 173...
## $ commonName       <chr> "Black-bellied Whistling-Duck", "Fulvous Whis...
## $ frenchCommonName <chr> "Dendrocygne Ã  ventre noir", "Dendrocygne fau...
## $ scientificName   <chr> "Dendrocygna autumnalis", "Dendrocygna bicolo...
## $ order            <chr> "Anseriformes", "Anseriformes", "Anseriformes...
## $ family           <chr> "Anatidae", "Anatidae", "Anatidae", "Anatidae...
## $ genus            <chr> "Dendrocygna", "Dendrocygna", "Anser", "Anser...
## $ species          <chr> "autumnalis", "bicolor", "canagicus", "caerul...
```

Subset by species AOU # (e.g. House Sparrow aou = 06882)

```
subset_speciesList(myData = flBBS, aou.ind = 06882) %>% glimpse()
```

```
## Observations: 138,393
## Variables: 23
## $ routedataid       <int> 6234830, 6234830, 6234830, 6234830, 6234830,...
## $ countrynum        <int> 840, 840, 840, 840, 840, 840, 840, 840, 840,...
## $ statenum          <int> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, ...
## $ route             <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ rpid              <int> 101, 101, 101, 101, 101, 101, 101, 101, 101,...
## $ year              <int> 1967, 1967, 1967, 1967, 1967, 1967, 1967, 19...
## $ aou               <int> 1840, 2000, 2010, 2890, 3100, 3131, 3160, 33...
## $ count10           <dbl> 0, 0, 0, 12, 1, 0, 3, 0, 0, 0, 0, 0, 0, 2, 2...
## $ count20           <dbl> 0, 0, 0, 22, 0, 0, 2, 0, 0, 0, 0, 0, 1, 7, 0...
## $ count30           <dbl> 1, 0, 0, 7, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,...
## $ count40           <dbl> 30, 3, 1, 7, 0, 1, 2, 1, 0, 0, 0, 1, 0, 6, 2...
## $ count50           <dbl> 5, 0, 0, 12, 0, 0, 0, 0, 1, 0, 0, 0, 2, 5, 1...
## $ stoptotal         <dbl> 3, 1, 1, 37, 1, 1, 7, 1, 1, 1, 1, 1, 2, 11, ...
## $ speciestotal      <dbl> 36, 3, 1, 60, 1, 1, 7, 1, 1, 1, 1, 1, 3, 20,...
## $ routeID           <chr> "25 1", "25 1", "25 1", "25 1", "25 1", "25 ...
## $ routename         <fct> OAK GROVE, OAK GROVE, OAK GROVE, OAK GROVE, ...
## $ active            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ latitude          <dbl> 30.92918, 30.92918, 30.92918, 30.92918, 30.9...
## $ longitude         <dbl> -87.40794, -87.40794, -87.40794, -87.40794, ...
## $ stratum           <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,...
## $ bcr               <int> 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, ...
## $ routetypeid       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ routetypedetailid <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
```

We could merge the bbs count data with the species list to avoid having to refer to AOU, then just subset using species name (e.g. 'House Sparrow).

```
flBBS <- left_join(flBBS, spp)
hospBBS <- flBBS %>% filter(commonName=="House Sparrow")
```

We can also use the `subset_SpeciesList` as a convenient way to **remove** taxonomic groups from the BBS data.

```
flBBS.subset <- subset_speciesList(flBBS, fam.ind = "Passeridae")
flBBS.subset <- subset_speciesList(flBBS, fam.ind = c("Passeridae", "Parulidae")) # or remove multiple
```

Retrieve BBS analysis results and data credibility measures

There are a few options for obtaining species trends estimates and credibility measures: 1) download the entire region-species csvs for various analyses or 2) provide a URl to species- or region-specific estimates for the 1966-2015 trend estimates.

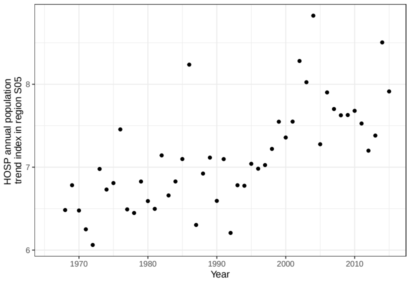
Option 1: Download CSV for all species-region combinations

The function `get_analysis_results` allows you to specify an analysis type, and upload all species-regions combination estimates or annual indices to object. Let's look at Florida House Sparrow trend estimates for Florida:

```
results <- get_analysis_results(analysis = "trend.ests") # default here is to obtain the 1966-2015 spec
results.flHOSP <- results %>% filter(Species.Name=="House Sparrow", Region.Code=="FLA")
```

Get annual trend estimates for Florida 1966-2016 analysis:

```
results <- get_analysis_results(analysis = "annual.inds.2016") # default here is to obtain the 1966-201
```



Options 2: Retrieve region-specific estimates using web-scraping

Another useful feature of this package is the ability to retrieve data credibility and species trend estimates from the BBS results using the function `get_credibility_trends`. This function allows the user to input a url to the region- or species-specific results page (see instructions below), as opposed to using function `get_analysis_results`. As an example, we retrieve the credibility scores and species trend estimates for **House Sparrows in Florida**.

```
cred <- get_credibility_trends() # default here is Florida House Sparrows.
```

```
# credibility colors correspond with the color scheme used on the BBS results page
cred %>% distinct(credibilityNumber, credibilityColor, credibilityClass)
```

```
##   credibilityNumber credibilityColor     credibilityClass
## 1                 2              Red important_deficiency
## 2                 0             Blue        no_deficiency
## 3                 1           Yellow           deficiency
```

Trend estimates are also listed in `cred` for Florida House Sparrow data:

```
## Observations: 1
## Variables: 12
## $ Species           <chr> "House Sparrow"
## $ N                 <int> 83
## $ Trend_1966_2015   <dbl> -7.08
## $ CI_2.5_1966_2015  <chr> "-7.99"
## $ CI_97.5_1966_2015 <chr> "-6.10"
## $ Trend_2005_2015   <dbl> -8.25
## $ CI_2.5_2005_2015  <chr> "-11.30"
## $ CI_97.5_2005_2015 <chr> "-5.28"
## $ RA                <dbl> 6.44
## $ credibilityNumber <chr> "0"
## $ credibilityColor  <fct> Blue
## $ credibilityClass  <fct> no_deficiency
```

Steps for obtaining argument "url" in function `get_credibility_trends`:

First, visit the USGS Patuxent Wildlife Research Center's website for BBS results Online https://www.mbr-pwrc.usgs.gov/.

Next, enable the drop-down **Survey Results**, and choose **Trend Estimates (by region)** (left) and choose the desired region (e.g. Florida). Finally, copy the URL address for the page including the results and credibility measures (e.g. Florida):

Acknowledgments

We thank the participatory scientists who collect data annually for the North American Breeding Bird Survey, and the Patuxent Wildlife Research Center for making these data publicly and easily accessible. Some functions in this package were adapted from the rBBS package and are mentioned in function source code as applicable.