

Package ‘birdproofr’

December 31, 2018

Title Bird Banding Data Validator

Version 1.0.0

Description birdproofr is a package of R tools for bird banding data validation under a set of rules written by Heidi Ware Carlisle, Intermountain Bird Observatory. The validator can be ran as a Shiny app for convenience, which includes utilities for viewing and downloading flagged data. Individual attributes can also be validated through function calls from the R console - please see IBO ruleset. The current birdproofr build has been updated for Fall 2018 banding. Support for a hummingbird ruleset is planned.

Depends R (>= 3.5.0), shiny (>= 1.2.0), dplyr (>= 0.7.5)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

run_birdproofr_app	2
validate_age	3
validate_age_bp_cp	3
validate_age_ffmlt	4
validate_age_ffwear	4
validate_age_ha	5
validate_age_hs	5
validate_age_skull	6
validate_all_list	6
validate_bandcode	7
validate_bandcode_species	7
validate_bandsize	8
validate_bandsize_disp	8
validate_bmlt	9
validate_bp	9
validate_bp_hs	10
validate_capttime	10
validate_cp	11
validate_cp_hs	11
validate_day	12
validate_day_species	12

validate_disp	13
validate_disp_status	13
validate_ey	14
validate_fat	14
validate_ffmolt	15
validate_ffwear	15
validate_ha_ffmolt	16
validate_ha_ffwear	16
validate_ha_ha2	17
validate_ha_skull	17
validate_hs_hs2	18
validate_location	18
validate_month	19
validate_month_species	19
validate_muscle	20
validate_net	20
validate_notes	21
validate_parasites	21
validate_sex	22
validate_sex_hs	22
validate_species	23
validate_status	23
validate_status_500	24
validate_tail	24
validate_weight	25
validate_wing	25
validate_year	26
validate_year_species	26

Index	27
--------------	-----------

run_birdproofr_app	<i>Runs birdproofr Shiny app</i>
--------------------	----------------------------------

Description

Runs birdproofr Shiny app

Usage

run_birdproofr_app()

validate_age	<i>Validate age column. Acceptable ages are: 0,1,2,4,5,6 –flag any records with blank age</i>
--------------	---

Description

Validate age column. Acceptable ages are: 0,1,2,4,5,6 –flag any records with blank age

Usage

```
validate_age(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with age issues

validate_age_bp_cp	<i>Check that age and BP/CP match. Age 2, 4, and 0 should always have 0 for both BP and CP</i>
--------------------	--

Description

Check that age and BP/CP match. Age 2, 4, and 0 should always have 0 for both BP and CP

Usage

```
validate_age_bp_cp(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with age/BP/CP issues

validate_age_ffmolt	<i>Validate age-ffmolt combinations. Blanks are okay, and can match with any age. Refer to table on rules page</i>
---------------------	--

Description

Validate age-ffmolt combinations. Blanks are okay, and can match with any age. Refer to table on rules page

Usage

```
validate_age_ffmolt(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with age/ffmolt issues

validate_age_ffwear	<i>Validate age-ffwear combinations.</i>
---------------------	--

Description

0 or 1 FF wear is highly suspicious for age 5 and 6. Flag all these records Sometimes 0 FF wear is normal if paired with S FF molt, but then micro-ageing is suspect, so we should flag the record either way, maybe with a message FF wear and age combination unlikely. Check this record

Usage

```
validate_age_ffwear(df)
```

Arguments

df	bird data frame
----	-----------------

Details

2+ FF wear is suspicious for age 4—add message age and FF wear combination unlikely

4+ is suspicious for age 2—add unlikely message

Value

data frame of rows with age/ffwear issues

validate_age_ha	<i>Validate age-how aged combinations</i>
-----------------	---

Description

Validate age-how aged combinations

Usage

```
validate_age_ha(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with age/how aged issues

validate_age_hs	<i>Validate age-how sexed combinations</i>
-----------------	--

Description

Validate age-how sexed combinations

Usage

```
validate_age_hs(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with age/how sexed issues

validate_age_skull	<i>Validate age and skull combinations. Allowable values for skull 0-6, 8,9, blank. Flag all values in the skull column that don't match these</i>
--------------------	--

Description

Validate age and skull combinations. Allowable values for skull 0-6, 8,9, blank. Flag all values in the skull column that don't match these

Usage

```
validate_age_skull(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with age/skull issues

validate_all_list	<i>Validate all columns, then store issues as a list of data frames</i>
-------------------	---

Description

Validate all columns, then store issues as a list of data frames

Usage

```
validate_all_list(df)
```

Arguments

df	bird data frame
----	-----------------

Value

list of issue data frames

validate_bandcode	<i>Validate band code. Make sure there are no blanks. Make sure the only values used are 1,R,4,5,8,R,U.</i>
-------------------	---

Description

Validate band code. Make sure there are no blanks. Make sure the only values used are 1,R,4,5,8,R,U.

Usage

```
validate_bandcode(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with band code issues

validate_bandcode_species	<i>Validate band code-species combinations. Make sure 4 and 8 are only used for species codes BADE and BALO</i>
---------------------------	---

Description

Validate band code-species combinations. Make sure 4 and 8 are only used for species codes BADE and BALO

Usage

```
validate_bandcode_species(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with band code/species issues

validate_bandsize	<i>Validate band size. Make sure there are no blanks. Make sure the only values used are 0A, 0, 1, 1B, 1A, 1C, 2, 3, 3A, 3B</i>
-------------------	---

Description

Validate band size. Make sure there are no blanks. Make sure the only values used are 0A, 0, 1, 1B, 1A, 1C, 2, 3, 3A, 3B

Usage

```
validate_bandsize(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with band size issues

validate_bandsize_disp	<i>Validate band size-disp combinations</i>
------------------------	---

Description

Validate band size-disp combinations

Usage

```
validate_bandsize_disp(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with band size/disp issues

validate_bmlt	<i>Validate body molt. Allowable values: 0-4, blank</i>
---------------	---

Description

Validate body molt. Allowable values: 0-4, blank

Usage

```
validate_bmlt(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with body molt issues

validate_bp	<i>Validate BP (0-5, blank okay)</i>
-------------	--------------------------------------

Description

Validate BP (0-5, blank okay)

Usage

```
validate_bp(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with BP issues

validate_bp_hs	<i>Validate how sexed and BP for females. If sexed by BP, BP value cannot be blank or 0</i>
----------------	---

Description

Validate how sexed and BP for females. If sexed by BP, BP value cannot be blank or 0

Usage

```
validate_bp_hs(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with BP/how sexed issues for females

validate_captime	<i>Validate cap time. Allowed values include: 650 to 1300. Flag all other values. Other values may happen only if there is a note, sometimes songbirds are caught during owls, hawk trapping, etc. All values should end in 0's</i>
------------------	---

Description

Validate cap time. Allowed values include: 650 to 1300. Flag all other values. Other values may happen only if there is a note, sometimes songbirds are caught during owls, hawk trapping, etc. All values should end in 0's

Usage

```
validate_captime(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with cap time issues

validate_cp	<i>Validate CP (0-3 allowed, blank okay)</i>
-------------	--

Description

Validate CP (0-3 allowed, blank okay)

Usage

```
validate_cp(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with CP issues

validate_cp_hs	<i>Validate how sexed and CP for males. If sexed by CL, CP value cannot be blank, 0, or 1 (i.e. CP must = 2 or 3)</i>
----------------	---

Description

Validate how sexed and CP for males. If sexed by CL, CP value cannot be blank, 0, or 1 (i.e. CP must = 2 or 3)

Usage

```
validate_cp_hs(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with CP/how sexed issues for males

validate_day	<i>Validate day. Valid: 1-31. no blanks except for BADE/BALO</i>
--------------	--

Description

Validate day. Valid: 1-31. no blanks except for BADE/BALO

Usage

```
validate_day(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with day issues

validate_day_species	<i>Validate day-species combinations</i>
----------------------	--

Description

Validate day-species combinations

Usage

```
validate_day_species(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with day/species issues

validate_disp	<i>Validate disp. Allowable values include: M,O,I,S,E,D,T,W,B,L,P, blank</i>
---------------	--

Description

Validate disp. Allowable values include: M,O,I,S,E,D,T,W,B,L,P, blank

Usage

```
validate_disp(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with disp issues

validate_disp_status	<i>Validate disp-status combinations. Any bird with a letter in disp should have a note explaining why and the status should say 500</i>
----------------------	--

Description

Validate disp-status combinations. Any bird with a letter in disp should have a note explaining why and the status should say 500

Usage

```
validate_disp_status(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with disp/status issues

validate_ey	<i>Validate EY in how aged. EY in the How Aged columns should only be used for species codes SPTO, DOWO, NOFL, RSFL, HAWO, DEJU, ORJU, SCJU, UDEJ –flag any other species that use this with note, Check in Pyle to confirm that this species can be aged by eye color</i>
-------------	--

Description

Validate EY in how aged. EY in the How Aged columns should only be used for species codes SPTO, DOWO, NOFL, RSFL, HAWO, DEJU, ORJU, SCJU, UDEJ –flag any other species that use this with note, Check in Pyle to confirm that this species can be aged by eye color

Usage

```
validate_ey(df)
```

Arguments

df bird data frame

Value

data frame of rows with EY issues

validate_fat	<i>Validate fat 0-5, blank are allowed. 6 fat is okay but only if there's a note</i>
--------------	--

Description

Validate fat 0-5, blank are allowed. 6 fat is okay but only if there's a note

Usage

```
validate_fat(df)
```

Arguments

df bird data frame

Value

data frame of rows with fat issues

validate_ffmolt	<i>Validate flight feather molt. Allowable values: N, S, J, A, blank</i>
-----------------	--

Description

Validate flight feather molt. Allowable values: N, S, J, A, blank

Usage

```
validate_ffmolt(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with ffmolt issues

validate_ffwear	<i>Validate flight feather wear. Allowable values: 0-5, blank</i>
-----------------	---

Description

Validate flight feather wear. Allowable values: 0-5, blank

Usage

```
validate_ffwear(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with ffwear issues

validate_ha_ffmolt	<i>Validate how aged-ffmolt combinations. If "how aged" says MR, FF molt must be S or J (can't be blank, N, or A)</i>
--------------------	---

Description

Validate how aged-ffmolt combinations. If "how aged" says MR, FF molt must be S or J (can't be blank, N, or A)

Usage

```
validate_ha_ffmolt(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with how aged/ffmolt issues

validate_ha_ffwear	<i>Validate how aged-ffwear combinations. If "how aged" says FF then FF Wear cannot be blank</i>
--------------------	--

Description

Validate how aged-ffwear combinations. If "how aged" says FF then FF Wear cannot be blank

Usage

```
validate_ha_ffwear(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with how aged/ffwear issues

validate_ha_ha2	<i>Validate how aged-how aged 2 combinations</i>
-----------------	--

Description

Validate how aged-how aged 2 combinations

Usage

```
validate_ha_ha2(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with ha/ha2 issues

validate_ha_skull	<i>Validate how aged and skull combinations</i>
-------------------	---

Description

Validate how aged and skull combinations

Usage

```
validate_ha_skull(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with how aged/skull issues

validate_hs_hs2	<i>Validate how sexed-how sexed 2 combinations</i>
-----------------	--

Description

Validate how sexed-how sexed 2 combinations

Usage

```
validate_hs_hs2(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with hs/hs2 issues

validate_location	<i>Validate location. Make sure there are no blanks</i>
-------------------	---

Description

Validate location. Make sure there are no blanks

Usage

```
validate_location(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with location issues

validate_month	<i>Validate month. Valid: 2-11. No blanks except for BADE BALO</i>
----------------	--

Description

Validate month. Valid: 2-11. No blanks except for BADE BALO

Usage

```
validate_month(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with month issues

validate_month_species	<i>Validate month-species combinations</i>
------------------------	--

Description

Validate month-species combinations

Usage

```
validate_month_species(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with month/species issues

validate_muscle	<i>Validate muscle. 2.5,3,4,5, blank allowed. 1 or 2 are allowed but MUST have a note, otherwise it's likely a type-o (check hard copy)</i>
-----------------	---

Description

Validate muscle. 2.5,3,4,5, blank allowed. 1 or 2 are allowed but MUST have a note, otherwise it's likely a type-o (check hard copy)

Usage

```
validate_muscle(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with muscle issues

validate_net	<i>Validate net. Allowable values: 1-12, blank. Some exceptions allowed with a note, e.g. owl nets but we should flag those exceptions anyway to make sure someone checks them</i>
--------------	--

Description

Validate net. Allowable values: 1-12, blank. Some exceptions allowed with a note, e.g. owl nets but we should flag those exceptions anyway to make sure someone checks them

Usage

```
validate_net(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with net issues

validate_notes	<i>Validate notes. Check that notes that mention either, FF, flat flies, or mites, lice, louse, mite, fly have a Y for parasite column</i>
----------------	--

Description

Validate notes. Check that notes that mention either, FF, flat flies, or mites, lice, louse, mite, fly have a Y for parasite column

Usage

```
validate_notes(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with notes issues

validate_parasites	<i>Validate parasites. If there is a Y in the parasites column there needs to be a note</i>
--------------------	---

Description

Validate parasites. If there is a Y in the parasites column there needs to be a note

Usage

```
validate_parasites(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with parasite column issues

validate_sex	<i>Validate sex column. Acceptable values= M F U–flag all the blanks</i>
--------------	--

Description

Validate sex column. Acceptable values= M F U–flag all the blanks

Usage

```
validate_sex(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with sex issues

validate_sex_hs	<i>Validate how sexed and sex combinations. Allowable values include: PL, EY,FF,MB,PC,LP,NL,MR,SK,TS, (blank only in second field, or for age 0)</i>
-----------------	--

Description

F: PL,BP,WL–first HS field can NOT be blank

Usage

```
validate_sex_hs(df)
```

Arguments

df	bird data frame
----	-----------------

Details

M: PL,CL,WL–first HS field can NOT be blank

U: always blank, or IC, If not blank, check hard copy for errors or white-out. If sex is whited out, leave as U. Check fields above and below to make sure there's not a data entry error

Value

data frame of rows with hs/sex issues

validate_species	<i>Validate species column. Refer to master species list to update</i>
------------------	--

Description

Validate species column. Refer to master species list to update

Usage

```
validate_species(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with species issues

validate_status	<i>Validate status. Allowable values for new bands: 300, 500. Blank is NOT valid</i>
-----------------	--

Description

Validate status. Allowable values for new bands: 300, 500. Blank is NOT valid

Usage

```
validate_status(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with status issues

validate_status_500	<i>Validate status 500s. ALL status 500's MUST have text in the note column and a letter in the disp column i.e. Note and Disp columns cannot be blank</i>
---------------------	--

Description

Validate status 500s. ALL status 500's MUST have text in the note column and a letter in the disp column i.e. Note and Disp columns cannot be blank

Usage

```
validate_status_500(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with status 500 issues

validate_tail	<i>Validate tail. Check if tail is below 30 or above 200</i>
---------------	--

Description

Validate tail. Check if tail is below 30 or above 200

Usage

```
validate_tail(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with tail issues

validate_weight	<i>Validate weight. Flag anything under 5 but GCKI or BCHU RUHU CAHU okay or over 200 raptors would be a rare exception</i>
-----------------	---

Description

Validate weight. Flag anything under 5 but GCKI or BCHU RUHU CAHU okay or over 200 raptors would be a rare exception

Usage

```
validate_weight(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with weight issues

validate_wing	<i>Validate wing. Check if wing is below 30 or above 200</i>
---------------	--

Description

Validate wing. Check if wing is below 30 or above 200

Usage

```
validate_wing(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with wing issues

validate_year	<i>Validate year. No blanks. Allowable values are any valid year between 1997 and current year except BADE BALO</i>
---------------	---

Description

Validate year. No blanks. Allowable values are any valid year between 1997 and current year except BADE BALO

Usage

```
validate_year(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with year issues

validate_year_species	<i>Validate year-species combinations</i>
-----------------------	---

Description

Validate year-species combinations

Usage

```
validate_year_species(df)
```

Arguments

df	bird data frame
----	-----------------

Value

data frame of rows with year/species issues

Index

run_birdproofr_app, 2

validate_age, 3

validate_age_bp_cp, 3

validate_age_ffmlt, 4

validate_age_ffwear, 4

validate_age_ha, 5

validate_age_hs, 5

validate_age_skull, 6

validate_all_list, 6

validate_bandcode, 7

validate_bandcode_species, 7

validate_bandsize, 8

validate_bandsize_disp, 8

validate_bmlt, 9

validate_bp, 9

validate_bp_hs, 10

validate_capttime, 10

validate_cp, 11

validate_cp_hs, 11

validate_day, 12

validate_day_species, 12

validate_disp, 13

validate_disp_status, 13

validate_ey, 14

validate_fat, 14

validate_ffmolt, 15

validate_ffwear, 15

validate_ha_ffmlt, 16

validate_ha_ffwear, 16

validate_ha_ha2, 17

validate_ha_skull, 17

validate_hs_hs2, 18

validate_location, 18

validate_month, 19

validate_month_species, 19

validate_muscle, 20

validate_net, 20

validate_notes, 21

validate_parasites, 21

validate_sex, 22

validate_sex_hs, 22

validate_species, 23

validate_status, 23

validate_status_500, 24

validate_tail, 24

validate_weight, 25

validate_wing, 25

validate_year, 26

validate_year_species, 26