

Package ‘vectorsurvR’

June 2, 2025

Type Package

Title Data Access and Analytical Tools for 'VectorSurv' Users

Version 1.4.0

Description Allows registered 'VectorSurv' <<https://vectorsurv.org/>> users access to data through the 'VectorSurv API' <<https://api.vectorsurv.org/>>. Additionally provides functions for analysis and visualization.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports rstudioapi, jsonlite, kableExtra, knitr, lubridate, stringr,
httr2, tidyr, magrittr, DT, purrr, sf

Suggests testthat (>= 3.0.0), rmarkdown, devtools,

VignetteBuilder knitr

Config/testthat/edition 3

Depends dplyr, R (>= 2.10)

NeedsCompilation no

Author Christina De Cesaris [aut, cre]

Maintainer Christina De Cesaris <cmdecesaris@ucdavis.edu>

Repository CRAN

Date/Publication 2025-06-02 20:40:02 UTC

Contents

getAbundance	2
getAbundanceAnomaly	3
getAbundanceMapData	4
getArthroCollections	5
getInfectionRate	6
getPools	7

getPoolsComparisionTable	8
getRegions	8
getSites	9
getSpatialFeatures	9
getToken	10
getVectorIndex	10
processAbunAnom	12
sample_collections	12
sample_pools	13
sample_spatial	14

Index	16
--------------	-----------

getAbundance	<i>Calculate abundance</i>
--------------	----------------------------

Description

Calculates abundance

Usage

```
getAbundance(  
  collections,  
  interval,  
  agency = NULL,  
  species = NULL,  
  trap = NULL,  
  sex = "female",  
  trapnight_min = 1,  
  trapnight_max = NULL,  
  separate_by = NULL  
)
```

Arguments

collections	Collections data retrieved from getArthroCollections()
interval	Calculation interval for abundance, accepts “collection_date”, “Biweek”, “Week”, and “Month.
agency	An optional vector for filtering agency by character code
species	An optional vector for filtering species. Species_display_name is the accepted notation. To see a list of species present in your data run unique(collections\$species_display_name). If species is unspecified, the default NULL will return data for all species in data.
trap	An optional vector for filtering trap type by acronym. Trap_acronym is the accepted notation. Run unique(collections\$trap_acronym) to see trap types present in your data. If trap is unspecified, the default NULL will return data for all trap types.

sex	An optional vector for filtering sex type. Accepts 'male', 'female', or 'other'. If sex is unspecified, the default NULL will return data for female sex.
trapnight_min	Minimum trap night restriction for calculation. Default is 1.
trapnight_max	Maximum trap night restriction for calculation. Default is no restriction.
separate_by	Separate/group the calculation by 'trap', 'species' or 'agency'. Default NULL does not separate.

Value

A dataframe of abundance calculations.

Examples

```
getAbundance(sample_collections,
              interval = 'Week',
              species = list('Cx pipiens'),
              trap = list('GRVD', 'C02'),
              sex = list("female"),
              trapnight_min = 1,
              trapnight_max = 5,
              separate_by = "species")
```

getAbundanceAnomaly	<i>Get Abundance Anomaly</i>
---------------------	------------------------------

Description

'getAbundanceAnomaly(...)' requires at least five years prior to the target_year of arthro collections data to calculate for the specified parameters. The function uses the methods of the Gateway Abundance Anomaly calculator, and will not work if there is fewer than five years of data present.

Usage

```
getAbundanceAnomaly(
  collections,
  interval,
  target_year,
  agency = NULL,
  species = NULL,
  trap = NULL,
  sex = "female",
  trapnight_min = 1,
  trapnight_max = NULL,
  separate_by = NULL
)
```

Arguments

collections	Collections data retrieved from 'getArthroCollections()'
interval	Calculation interval for abundance, accepts "collection_date", "Biweek", "Week", and "Month"
target_year	Year to calculate analysis on. Collections data must have a year range of at least (target_year - 5, target_year)
agency	An optional vector for filtering agency by character code
species	An optional vector for filtering species. Species_display_name is the accepted notation. To see a list of species present in your data run unique(collections\$species_display_name). If species is unspecified, the default NULL will return data for all species in data.
trap	An optional vector for filtering trap type by acronym. Trap_acronym is the accepted notation. Run unique(collections\$trap_acronym) to see trap types present in your data. If trap is unspecified, the default NULL will return data for all trap types.
sex	An optional vector for filtering sex type. Accepts 'male', 'female', or 'other'. If sex is unspecified, the default NULL will return data for female sex.
trapnight_min	Minimum trap night restriction for calculation. Default is 1.
trapnight_max	Maximum trap night restriction for calculation. Default is no restriction.
separate_by	Separate/group the calculation by 'trap', 'species', 'agency' or 'county'. Default NULL does not separate.

Value

Abundance anomaly calculation

Examples

```
getAbundanceAnomaly(sample_collections, "Biweek", target_year=2020, species="Cx pipiens")
```

getAbundanceMapData	<i>Get Abundance Map Data</i>
---------------------	-------------------------------

Description

'getAbundanceMapData()' retrieves spatial feature data and associates it with collection locations.

Usage

```
getAbundanceMapData(collections, spatial, interval, selected_features)
```

Arguments

collections	A dataframe containing collection coordinates
spatial	A dataframe containing spatial data corresponding to the collections
interval	Calculation interval for abundance, accepts “Biweek”, “Week”, and “Month”
selected_features	A character vector of spatial feature names to filter by

Value

A dataframe of collections with associated spatial regions

getArthroCollections *Get arthropod collections data*

Description

‘getArthroCollections()’ obtains collections data on a year range [start_year, end_year] for authorized VectorSurv Gateway accounts.

Usage

```
getArthroCollections(
  token,
  start_year,
  end_year,
  arthropod,
  agency_ids = NULL,
  spatial_features = NULL
)
```

Arguments

token	A valid access token returned from ‘getToken()’
start_year	Start year of data
end_year	End year of data
arthropod	Specify arthropod type from: ‘mosquito’, ‘tick’
agency_ids	Filter on agency id, default to NULL for all available agencies, otherwise provide a vector of agency ids, such as ‘agency_ids = c(55,56)’
spatial_features	Filter data by spatial feature

Value

A dataframe of collections data

Examples

```
## Not run:
token = getToken()
collections = getArthroCollections(token, 2021, 2022, 'mosquito',c(55,56), TRUE)
## End(Not run)
```

getInfectionRate	<i>Calculate infection rate</i>
------------------	---------------------------------

Description

‘getInfectionRate()’ requires at least five years prior to the target_year of arthro collections data to calculate for the specified parameters. The function uses the methods of the Gateway Abundance Anomaly calculator, and will not work if there is fewer than five years of data present.

Usage

```
getInfectionRate(
  pools,
  interval,
  target_disease,
  pt_estimate,
  scale = 1000,
  agency = NULL,
  species = NULL,
  trap = NULL,
  sex = "female",
  separate_by = NULL,
  wide = FALSE
)
```

Arguments

pools	Pools data retrieved from ‘getPools()’
interval	Calculation interval for infection rate, accepts “collection_date”, “Biweek”, “Week”, and “Month”
target_disease	The disease to calculate infection rate for—i.e. “WNV”. Disease acronyms are the accepted input. To see a list of disease acronyms, run ‘unique(pools\$test_target_acronym)’
pt_estimate	The estimation type for infection rate. Options include: “mle”, “bc-mle”, “mir”
scale	Constant to multiply infection rate by
agency	An optional vector for filtering agency by character code
species	An optional vector for filtering species. Species_display_name is the accepted notation. To see a list of species present in your data run unique(collections\$species_display_name). If species is unspecified, the default NULL will return data for all species in data.

trap	An optional vector for filtering trap type by acronym. Trap_acronym is the is the accepted notation. Run <code>unique(collections\$trap_acronym)</code> to see trap types present in your data. If trap is unspecified, the default NULL will return data for all trap types.
sex	An optional vector for filtering sex type. Accepts 'male', 'female', or 'other'. If sex is unspecified, the default NULL will return data for female sex.
separate_by	Separate/group the calculation by 'trap', 'species' or 'agency'. Default NULL does not separate.
wide	Should the data be returned in wide/spreadsheet format

Value

Dataframe of infection rate calculation

getPools	<i>Get Pools data</i>
----------	-----------------------

Description

Retrieves VectorSurv pools data for desired year range

Usage

```
getPools(token, start_year, end_year, arthropod, agency_ids = NULL)
```

Arguments

token	access token retrieved from 'getToken()'
start_year	Beginning of year range
end_year	End of year range
arthropod	Specify arthropod type from: 'mosquito', 'tick', 'nontick'
agency_ids	Filter on agency id, default to NULL for all available agencies, otherwise provide a vector of agency ids

Value

Dataframe of pools data

Examples

```
## Not run:
token = getToken()
getPools(token, start_year = 2020, end_year = 2021, arthropod = 'tick', 55)
## End(Not run)
```

getPoolsComparisionTable

Get Pools Frequency Table

Description

‘getPoolsComparisionTable()’ produces a frequency table for positive, negative, and pending pools counts by year and species. The more years present in the data, the larger the table.

Usage

```
getPoolsComparisionTable(pools, interval, target_disease, separate_by = NULL)
```

Arguments

pools	Pools data retrieved from ‘getPools()’
interval	Calculation interval for comparison table, accepts “collection_date”, “Biweek”, “Week”, and “Month
target_disease	The disease to calculate infection rate for–i.e. “WNV”. Disease acronyms are the accepted input. To see a list of disease acronyms, run ‘unique(pools\$target_acronym)’
separate_by	Separate/group the calculation by ‘trap’, ‘species’ or ‘agency’. Default NULL does not separate.

Value

Frequency table of for pools data

Examples

```
getPoolsComparisionTable(sample_pools,
                           interval = "Biweek",
                           target_disease = "WNV",
                           separate_by = "species")
```

getRegions

Get region data

Description

‘getSites()’ obtains site data for authorized VectorSurv Gateway accounts.

Usage

```
getRegions(token)
```


Arguments

token A valid access token returned from 'getToken()'

Value

A dataframe of region data, used internally to merge spatial information to collections

getSites	<i>Get sites data</i>
----------	-----------------------

Description

'getSites()' obtains site data for authorized VectorSurv Gateway accounts.

Usage

```
getSites(token)
```

Arguments

token A valid access token returned from 'getToken()'

Value

A dataframe of site data

Examples

```
## Not run:  
token = getToken()  
sites = getSites(token)  
## End(Not run)
```

getSpatialFeatures	<i>Get Spatial Features data</i>
--------------------	----------------------------------

Description

Retrieves VectorSurv spatial features data with optional agency filtering

Usage

```
getSpatialFeatures(token, agency_ids = NULL)
```

Arguments

- token Access token retrieved from ‘getToken()’
- agency_ids Optional vector of agency IDs to filter by

Value

An sf object containing spatial features data

Examples

```
## Not run: token=getToken()
spatial = getSpatialFeatures(token)
## End(Not run)
```

getToken	<i>Get authentication token</i>
----------	---------------------------------

Description

getToken() returns a token needed to run getArthroCollections() and getPools(). Prints agencies associated with account credentials. The function prompts users for a VectorSurv account credentials.

Usage

```
getToken()
```

Value

User token

Examples

```
## Not run: token = getToken()
```

getVectorIndex	<i>Calculate vector index</i>
----------------	-------------------------------

Description

‘getVectorIndex()’requires at least five years prior to the target_year of arthro collections data to calculate for the specified parameters. The function uses the methods of the Gateway Abundance Anomaly calculator, and will not work if there is fewer than five years of data present.

Usage

```
getVectorIndex(
  collections,
  pools,
  interval,
  target_disease,
  pt_estimate,
  scale = 1000,
  agency = NULL,
  species = NULL,
  trap = NULL,
  sex = NULL,
  trapnight_min = 1,
  trapnight_max = NULL,
  separate_by = NULL,
  wide = FALSE
)
```

Arguments

<code>collections</code>	Collections data retrieved from ‘getArthroCollections()’
<code>pools</code>	Pools data retrieved from ‘getPools()’
<code>interval</code>	Calculation interval for vector index, accepts “collection_date”, “Biweek”, “Week”, and “Month”
<code>target_disease</code>	The disease to calculate infection rate for–i.e. “WNV”. Disease acronyms are the accepted input. To see a list of disease acronyms, run ‘unique(pools\$target_acronym)’
<code>pt_estimate</code>	The estimation type for infection rate. Options include: “mle”, “bc-”mle”, “mir”
<code>scale</code>	Constant to multiply infection rate, default is 1000
<code>agency</code>	An optional vector for filtering agency by character code
<code>species</code>	An optional vector for filtering species. Species_display_name is the accepted notation. To see a list of species present in your data run unique(collections\$species_display_name). If species is unspecified, the default NULL will return data for all species in data.
<code>trap</code>	An optional vector for filtering trap type by acronym. Trap_acronym is the accepted notation. Run unique(collections\$trap_acronym) to see trap types present in your data. If trap is unspecified, the default NULL will return data for all trap types.
<code>sex</code>	An optional vector for filtering sex type. Accepts ‘male’, ‘female’, or ‘other’. If sex is unspecified, the default NULL will return data for female sex.
<code>trapnight_min</code>	Minimum trap night restriction for calculation. Default is 1.
<code>trapnight_max</code>	Maximum trap night restriction for calculation. Default is no restriction.
<code>separate_by</code>	Separate/group the calculation by ‘trap’, ‘species’ or ‘agency’. Default NULL does not separate.
<code>wide</code>	Should the data be returned in wide/spreadsheet format

Value

Dataframe containing the vector index calculation

Examples

```
getVectorIndex(sample_collections, sample_pools, "Month", "WNV", "mle", wide = FALSE )
```

processAbunAnom	<i>Process abundance anomaly</i>
-----------------	----------------------------------

Description

‘processAbunAnom()’ processes the output returned from ‘getAbundanceAnomaly()’ into a long form suitable for plotting using ‘ggplot’

Usage

```
processAbunAnom(AbAnomOutput)
```

Arguments

AbAnomOutput output from ‘getAbunAnom()’

Value

Abundance anomaly output processed into long form, used for plotting functions

sample_collections	<i>Sample Mosquito Collections Data</i>
--------------------	---

Description

Sample Mosquito Collections data imitates the essential components of real mosquito collections data

Usage

```
sample_collections
```

Format

A data frame with 2500 rows and 13 variables:

agency_code character Four letter agency code
 agency_id integer Unique agency id number
 county character County which site resides
 collection_id double Collection identification number
 collection_date character The date the trap was picked up for collection
 num_trap integer The number of unique traps in operation at one site
 site_code integer Identifying code of site
 surv_year double Surveillance year of collection
 trap_nights integer The number of nights a trap was in the field
 trap_problem_bit logical If there was an issue with the trap
 num_count integer Number of arthropods present in collection
 sex_type character Sex of collected arthropods
 species_display_name character Species name of collected arthropods
 trap_acronym character The acronym of the trap placed in the field
 collection_longitude numeric longitude of collection
 collection_latitude numeric latitude of collection

Source

<https://vectorsurv.org/>

sample_pools

Sample Pools Data

Description

Sample Pools data imitates the essential components of real mosquito pools data needed for calculations

Usage

sample_pools

Format

A data frame with 2500 rows and 10 variables:

agency_code character Four letter agency code
 agency_id integer Unique agency id number
 id integer Pool identification number
 surv_year integer Surveillance year of pool
 site_code integer Identifying code of site
 collection_date character The date the trap was picked up for collection
 sex_type integer Sex type of collected arthropods
 num_count integer Number of arthropods present in collection
 test_target_acronym character The disease being tested for in the pool
 test_method_name character Method used to test pool for disease
 test_status_name character Status of the tested disease, confirmed or negative
 trap_acronym character The acronym of the trap placed in the field
 species_display_name character Species name of collected arthropods
 pool_longitude numeric longitude of pool
 pool_latitude numeric latitude of pool

Source

<https://vectorsurv.org/>

sample_spatial	<i>Sample Spatial Data</i>
----------------	----------------------------

Description

Sample Spatial data imitates spatial feature data

Usage

```
sample_spatial
```

Format

A data frame with 3 rows and 4 variables:

agency character Agency name
 agency_id integer Unique agency id number
 id integer ID of spatial feature
 name character Name of spatial feature
 geometry multipolygon shape geometry spatial feature

sample_spatial

15

Source

<https://vectorsurv.org/>

Index

- * **abundance**
 - getAbundanceAnomaly, [3](#)
- * **authentication**
 - getToken, [10](#)
- * **datasets**
 - sample_collections, [12](#)
 - sample_pools, [13](#)
 - sample_spatial, [14](#)
- * **infection**
 - getInfectionRate, [6](#)
- * **pools**
 - getInfectionRate, [6](#)
 - getPools, [7](#)
- * **rate**
 - getInfectionRate, [6](#)

[getAbundance, 2](#)
[getAbundanceAnomaly, 3](#)
[getAbundanceMapData, 4](#)
[getArthroCollections, 5](#)
[getInfectionRate, 6](#)
[getPools, 7](#)
[getPoolsComparisionTable, 8](#)
[getRegions, 8](#)
[getSites, 9](#)
[getSpatialFeatures, 9](#)
[getToken, 10](#)
[getVectorIndex, 10](#)

[processAbunAnom, 12](#)

[sample_collections, 12](#)
[sample_pools, 13](#)
[sample_spatial, 14](#)