Package 'primate'

October 14, 2022

Version 0.2.0

Date 2022-05-05

Title Tools and Methods for Primatological Data Science

Author David Schruth [aut][cre], Marc Myers [ctb], Noel Rowe [aut]

Maintainer David Schruth <data@anthropoidea.org>

Depends R (>= 1.8.0), caroline

Suggests RJDBC

Description Data from All the World's Primates relational SQL database and other tabular datasets are made available via drivers and connection functions. Additionally we provide several functions and examples to facilitate the merging and aggregation of these tabular inputs.

License Apache License

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-09 07:20:02 UTC

R topics documented:

add.gnsp.clmn *Add a genus species column*

Description

Adds a genus_species column to the specified dataframe

Usage

```
add.gnsp.clmn(df,gn="Genus",sp="Species",rownames=FALSE,new.col=TRUE, gnsp.col = "gn_sp")
```

Arguments

df	input data.frame
gn	column name for genus
sp	column name for species
rownames	use the new gn_sp column to assign data.frame rownames
new.col	TRUE if gn_sp column is to be retained, FALSE if it is to be removed
gnsp.col	the name of the new column to add, by default is "gn_sp"

Value

modified data.frame (with genus species info concatenated and added)

Examples

```
primates.tab <- AWP.read.pkg.tab(tab.nm='dbo_tblGrovesMonkeys', id.clmn='MonkeyNumberGroves')
primates.tab <- add.gnsp.clmn(primates.tab,gn="Genus",sp="Species",rownames=FALSE,new.col=TRUE)
```

AWP.connect *Create a connection to the SQL DB*

Description

Connect to the All the World's Primate Database

Usage

```
AWP.connect(drv=AWP.driver(), prefix='jdbc:jtds:sqlserver',
            server="s09.everleap.com", port=1433,
        db="DB_3918_atwpreview", user="DB_3918_atwpreview_user", pw="atwpreview_$_j")
```

Arguments

drv	driver (output from AWP.driver()
prefix	prefix to the URL (before "://")
server	domain name for the database server
port	port name used by the server's database
db	database name
user	database user name
pw	database user password

Value

a connection object for SQL

Examples

```
con <- AWP.connect(drv=AWP.driver())
```

AWP.driver *Load the driver to access the All the World's Primate remote SQL database*

Description

Load the driver to utilize the database sofware

Usage

```
AWP.driver(drv.name="net.sourceforge.jtds.jdbc.Driver",
           drv.file=system.file("drivers","jtds-1.2.8.jar", package='primate'))
```

Arguments

drv.name	The name of the driver
drv.file	The file name for the database driver

Value

driver argument to AWP.connect

Examples

```
AWP.driver()
```

AWP.get.lookup.table *Get a Lookup Table*

Description

Get the lookup table from the All the World's Primates SQL database

Usage

```
AWP.get.lookup.table(con=AWP.connect(),tab.nm="TextType")
```

Arguments

con	connection object
tab.nm	table name (for the parent table)

Value

a data.frame corresponding to SQL table

Examples

```
AWP.get.lookup.table(con=AWP.connect(),tab.nm="TextType")
```

AWP.get.SQL.table *Get a SQL Table*

Description

Retrieve a table from the All the World's Primates SQL database

Usage

```
AWP.get.SQL.table(con=AWP.connect(), tab.nm="tblGrovesMonkeys", clmns=c('all'),xpnd=FALSE)
```

Arguments

tab.nm	table name (defaults to the main primate species list)
con	connection object
clmns	columns to return
xpnd	expand the lookup column codes into full text strings

Value

a data.frame corresponding to SQL table

Examples

```
online.version <- AWP.get.SQL.table(tab.nm='LMType') #a small example table
```

`AWP.list.SQL.tables` *List the SQL tables*

Description

List available tables from the All the World's Primates SQL database

Usage

```
AWP.list.SQL.tables(con=AWP.connect(), all=FALSE)
```

Arguments

`con`	connection object from AWP.connect
`all`	list all tables available

Value

a list (vector) of SQL table names

Examples

```
AWP.list.SQL.tables(con=AWP.connect(), all=FALSE)
```

`AWP.read.pkg.tab` *Read a table from the All the World's primates example microarchive within this package*

Description

Read an All the World's Primates table from the local package cache.

Usage

```
AWP.read.pkg.tab(tab.nm='dbo_tblGrovesMonkeys', id.clmn=NA)
```

Arguments

`tab.nm`	table name
`id.clmn`	id column of table

Value

data.frame corresponding to SQL table

Examples

```
primates.tab <- AWP.read.pkg.tab(tab.nm='dbo_tblGrovesMonkeys', id.clmn='MonkeyNumberGroves')
```

`AWP.run.SQL` *Run SQL querries*

Description

Run arbitrary SQL querries on the All the World's Primate database

Usage

```
AWP.run.SQL(con=AWP.connect(), sql=NULL)
```

Arguments

`con`	connection object
`sql`	SQL string

Value

results of query

Examples

```
AWP.run.SQL(con=AWP.connect(), sql=NULL)
```

`regroup.equivalent` *Re-group data.frame by Genus_species in either old or new dataframe*

Description

Regroup based on the old or the new data frame using a direction parameter.

Usage

```
regroup.equivalent(df, gnsp.old, gnsp.new, clmns, agg='mean', direction='old2new')
```

Arguments

`df`	a dataframe
`gnsp.old`	old nomenclature
`gnsp.new`	new nomenclature
`clmns`	the columns in the data.frame to re-group
`agg`	the aggregation type
`direction`	the aggregation priority

Value

a regrouped data frame

Examples

```
primates.tab <- AWP.read.pkg.tab(tab.nm='dbo_tblGrovesMonkeys', id.clmn='MonkeyNumberGroves')
primates.tab <- add.gnsp.clmn(primates.tab,gn="Genus",sp="Species",rownames=FALSE,gnsp.col='gn_sp')
primates.tab <- add.gnsp.clmn(primates.tab,gn="Genus",sp="Species",rownames=FALSE,gnsp.col='gnspg')
pri.grpd <- regroup.gnsp(df=primates.tab,clmns=colnames(primates.tab), agg='max')
out <- regroup.equivalent(pri.grpd, gnsp.old=gn_sp, gnsp.new='gnspg',
                          clmns='MonkeyNumberGroves', agg='paste', direction='old2new')
```

regroup.gnsp *Re-group data.frame by Genus_species*

Description

Regroup a given data.frame by a column designated as unique genus_species combination. This function is essentially a wrapper for caroline:::groupBy()

Usage

```
regroup.gnsp(df,clmns,agg='mean',by='gn_sp')
```

Arguments

df	a dataframe
clmns	columns
agg	type of aggregation to be used
by	the column name by which the data.frame should be re-grouped

Value

returned value

Examples

```
primates.tab <- AWP.read.pkg.tab(tab.nm='dbo_tblGrovesMonkeys', id.clmn='MonkeyNumberGroves')
out <- regroup.gnsp(df=primates.tab,clmns=colnames(primates.tab), agg='paste')
```

updatevals *Update the values of an AWP data.frame*

Description

Update the values in an old dataframe with the values in a new dataframe. Useful for comparing a pre-existing or self-assembled dataset with AWP.

Usage

```
updatevals(x,y=NULL,v.old,v.new,verbose=TRUE,update=FALSE,
                    na.only=TRUE,all=TRUE,missing.only=TRUE)
```

Arguments

x	first dataframe
y	second dataframe
v.old	variable old
v.new	variable new
verbose	get all the messages
update	update all the old with everything new
na.only	just update the missing values in the old dataframe
all	perform merge on all columns
missing.only	update only those that have missing values

Value

values of one data frame are updated to reflect new data in another

Examples

```
pri.tab <- AWP.read.pkg.tab(tab.nm='Locomotion')
#pri.AWP <- AWP.get.SQL.table(tab.nm='Locomotion')

dim(pri.tab) #should may be fewer cols or rows locally ...
#dim(pri.AWP) # than there are available online.

apply(pri.tab, 2, function(x) sum(is.na(x))) # also more missing values
#apply(pri.AWP, 2, function(x) sum(is.na(x))) #  locally than online

# update the "Comment" column locally with the same online

vars <- c('LocomotionID','Comment')
#tmp <- merge(x=pri.tab[,c(vars)] ,y=pri.AWP[,c(vars)], by='LocomotionID')

#out <- updatevals(x=tmp,y=NULL,v.old='Comment.x',v.new='Comment.y')
```

Index