

Package : dabr

Corentin Bretonniere & Antoine Serreau

11/11/2020

Installation du package “dabr”

Le package “dabr” est utile en Data Management car il sert à gérer des bases de données, nous allons voir comment l’utiliser et quelles sont les fonctions qui le composent. Il est nécessaire d’également installer les packages suivant (à noter qu’il faut mettre “no” si des compilations sont demandées).

```
## Installing package into '/Users/corentinbretonniere/Library/R/3.5/library'
## (as 'lib' is unspecified)

## installing the source package 'dabr'

## Installing package into '/Users/corentinbretonniere/Library/R/3.5/library'
## (as 'lib' is unspecified)

##
##   There is a binary version available but the source version is later:
##       binary source needs_compilation
## remotes  2.1.1  2.2.0                FALSE

## installing the source package 'remotes'
##
## Installing package into '/Users/corentinbretonniere/Library/R/3.5/library'
## (as 'lib' is unspecified)

##
##   There is a binary version available but the source version is later:
##       binary source needs_compilation
## dplyr    0.8.5  1.0.2                TRUE

## installing the source package 'dplyr'
##
## Installing package into '/Users/corentinbretonniere/Library/R/3.5/library'
## (as 'lib' is unspecified)

##
##   There is a binary version available but the source version is later:
##       binary source needs_compilation
## RMariaDB 1.0.8 1.0.11                TRUE

## installing the source package 'RMariaDB'
```

```
##
## Attaching package: 'dabr'

## The following object is masked from 'package:stats':
##
##      update

## The following objects are masked from 'package:base':
##
##      attributes, quote
```

Une fois tous les packages installés et librairies ouverte nous allons devoir quitter R quelques temps !

Il nous faut télécharger mariadb qui est un moteur de base de données gratuit, il nous servira par la suite à convertir notre fichier CSV en Data base SQL.

- Ceux sous windows il faut télécharger via ce lien :

mariadb-10.5.8-winx64.msi .

- Ceux sur mac OS vous devez installer avec cette commande depuis le terminal :

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- Pour macOS comme pour linux, une fois dans votre terminal entrez cette commande :
brew install mariadb

puis : mysql.server start

puis : mysql -uroot -p

changez le mot de passe de votre serveur SQL avec la commande : sudo mysqladmin -u root -p password root

et lancer mariadb avec la commande : brew service start mariadb

avec la commande : "mysql" vous verrez que vous êtes bien admin de votre base de donnée

puis entrez la commande : FLUSH PRIVILEGES

Ensuite téléchargez <https://sequepro.com/download>

Une fois le téléchargement fait lancez sequepro, entrez les paramètres de votre serveur mySQL et mettez "localhost" à hôte.

Inserez votre base de données et paramétrez vos colonnes comme vous le souhaitez.

Une fois toutes ces démarches fastidieuses faites, nous pouvons revenir sur R et enfin pouvoir tâter le package "dabr". Pour savoir quelles fonctions sont incluses dans "dabr" il suffit d'entrer cette commande :

```
ls("package:dabr")
## [1] "attributes"      "close_conn"      "delete"          "get_attr"
## [5] "insert"          "is.connected"    "list_tables"     "open_conn_mysq
1"
## [9] "quote"           "select"          "select_all"      "update"
```

Dans la présentation du package dabr je vais utiliser une data base téléchargée sur kaggle avec des diverses données en relation avec les 50 musiques les plus écoutées sur Spotify en 2019 (Artiste, Nom de la musique, BPM,...).

1/ fonction “dbconnect”

Cette fonction permet de créer la connecter à la database préalablement crée à l’aide de mariadb sur Sequelpro. Dans R note database est appelée avec “conn” :

```
conn <- RMariaDB::dbConnect(RMariaDB::MariaDB(),
                             user = "root",
                             password = 'root',
                             dbname = 'Top50',
                             host = "127.0.0.1",
                             port = 3306)
```

2/ La fonction “is.connected”

Cette fonction sert à vérifier que la table est bien connectée.

```
is.connected.MariaDBConnection <- function(conn, ...) {
  RMariaDB::dbIsValid(conn)
}

is.connected.MariaDBConnection(conn)

## [1] TRUE
```

3/ fonction “close_conn”

Cette fonction permet, en opposition à dbconnect, de fermer la connection à une data base.

```
close_conn.MariaDBConnection <- function(conn, ...) {
  RMariaDB::dbDisconnect(conn)
}

close_conn.MariaDBConnection(conn)
```

Nous pouvons vérifier que la connexion à la dababase à bien été interrompu :

```
is.connected.MariaDBConnection(conn)

## [1] FALSE
```

Nous allons ré-établir la connexion pour parcourir les autres fonctions que contient le package “dabr”.

```
conn <- RMariaDB::dbConnect(RMariaDB::MariaDB(),
                             user = "root",
                             password = 'root',
                             dbname = 'Top50',
                             host = "127.0.0.1",
                             port = 3306)
```

```
is.connected.MariaDBConnection(conn)
```

```
## [1] TRUE
```

4/ La fonction “list_tables”

Cette fonction sert tout simplement à regarder quelles sont les colonnes de notre database

```
dabr::list_tables(conn)
```

```
##
##
## |top50      |
## |:-----|
## |id         |
## |Track Name |
## |Artist Name|
## |Genre      |
## |BPM        |
## |Energy %   |
## |Danceability|
## |Loudness dB|
## |Good mood  |
## |Duration   |
## |Popularity  |
## |Speechiness|
```

5/ fonction “delete”

Cette fonction sert à supprimer une requete (query).

Le code ci-dessous vas servir à enlever les espaces avant et après les data, s’il y en a.

```
query <- paste(unlist(lapply(c(), trimws)), collapse = " ")

delete.MariaDBConnection <- function(conn, ..., quiet = FALSE) {query}

dabr::delete(conn, query, quiet = FALSE)

## [1] ""
```

6/ fonction "select_all"

Cette commande permet de sélectionner toute la table en question :

```
dabr::select_all(conn, "top50")
```

```
## Executing:
```

```
## SELECT * FROM top50
```

```
##
```

```
## Results: 50 records were found.
```

```
##      id                                     Track Name
## 1    51                                     Señorita
## 2    52                                     Chin
## 3    53          boyfriend (with Social House
## 4    54          Beautiful People (feat. Khalid
## 5    55          Goodbyes (Feat. Young Thug
## 6    56          I Don't Care (with Justin Bieber
## 7    57                                     Ransom
## 8    58          How Do You Sleep
## 9    59          Old Town Road - Remix
## 10   60          bad guy
## 11   61          Callaita
## 12   62          Loco Contigo (feat. J. Balvin & Tyga
## 13   63          Someone You Love
## 14   64          Otro Trago - Remix
## 15   65          Money In The Grave (Drake ft. Rick Ross
## 16   66          No Guidance (feat. Drake
## 17   67          LA CANCIÓN
## 18   68          Sunflower - Spider-Man: Into the Spider-Vers
## 19   69          Lalala
```

a			
##	20	70	Truth Hurt
s			
##	21	71	Piece Of Your Hear
t			
##	22	72	Panin
i			
##	23	73	No Me Conoce - Remi
x			
##	24	74	Soltera - Remi
x			
##	25	75	bad guy (with Justin Bieber
)			
##	26	76	If I Can't Have Yo
u			
##	27	77	Dance Monke
y			
##	28	78	It's Yo
u			
##	29	79	Con Calm
a			
##	30	80	QUE PRETENDE
s			
##	31	81	Takeawa
y			
##	32	82	7 ring
s			
##	33	83	0.95833333333333
3			
##	34	84	The London (feat. J. Cole & Travis Scott
)			
##	35	85	Never Really Ove
r			
##	36	86	Summer Days (feat. Macklemore & Patrick Stump of Fall Out Boy
)			
##	37	87	Otro Trag
o			
##	38	88	Antisocial (with Travis Scott
)			
##	39	89	Sucke
r			
##	40	90	fuck, i'm lonely (with Anne-Marie) - from "13 Reasons Why: Season 3
”			
##	41	91	Higher Lov
e			
##	42	92	You Need To Calm Dow
n			
##	43	93	Shallo
w			
##	44	94	Tal

k									
##	45	95						Con Altur	
a									
##	46	96						One Thing Righ	
t									
##	47	97						Te Robar	
é									
##	48	98						Happie	
r									
##	49	99						Call You Min	
e									
##	50	100						Cross Me (feat. Chance the Rapper & PnB Rock	
)									
##			Artist Name	Genre	BPM	Energy %	Danceability	Loudness	dB
##	1		Shawn Mendes	canadian pop	191	55	76	-6	
##	2		Anuel AA	reggaeton flow	302	81	79	-4	
##	3		Ariana Grande	dance pop	186	80	40	-4	
##	4		Ed Sheeran	pop	198	65	64	-8	
##	5		Post Malone	dfw rap	175	65	58	-4	
##	6		Ed Sheeran	pop	220	68	80	-5	
##	7		Lil Tecca	trap music	131	64	75	-6	
##	8		Sam Smith	pop	202	68	48	-5	
##	9		Lil Nas X	country rap	157	62	88	-6	
##	10		Billie Eilish	electropop	194	43	70	-11	
##	11		Bad Bunny	reggaeton	251	62	61	-5	
##	12		DJ Snake	dance pop	185	71	82	-4	
##	13		Lewis Capaldi	pop	182	41	50	-6	
##	14		Sech	panamanian pop	288	79	73	-2	
##	15		Drake	canadian hip hop	205	50	83	-4	
##	16		Chris Brown	dance pop	261	45	70	-7	
##	17		J Balvin	latin	243	65	75	-6	
##	18		Post Malone	dfw rap	158	48	76	-6	
##	19		Y2K	canadian hip hop	161	39	84	-8	
##	20		Lizzo	escape room	173	62	72	-3	
##	21		MEDUZA	pop house	153	74	68	-7	
##	22		Lil Nas X	country rap	115	59	70	-6	
##	23		Jhay Cortez	reggaeton flow	309	79	81	-4	
##	24		Lunay	latin	266	78	80	-4	
##	25		Billie Eilish	electropop	195	45	67	-11	
##	26		Shawn Mendes	canadian pop	191	82	69	-4	
##	27		Tones and I	australian pop	210	59	82	-6	
##	28		Ali Gatie	canadian hip hop	213	46	73	-7	
##	29		Daddy Yankee	latin	193	86	74	-3	
##	30		J Balvin	latin	222	79	64	-4	
##	31		The Chainsmokers	edm	210	51	29	-8	
##	32		Ariana Grande	dance pop	179	32	78	-11	
##	33		Maluma	reggaeton	176	71	78	-5	
##	34		Young Thug	atl hip hop	200	59	80	-7	
##	35		Katy Perry	dance pop	224	88	77	-5	
##	36		Martin Garrix	big room	164	72	66	-7	

##	37	Sech	panamanian	pop	226	70	75	-5
##	38	Ed Sheeran		pop	162	82	72	-5
##	39	Jonas Brothers		boy band	181	73	84	-5
##	40	Lauv		dance pop	199	56	81	-6
##	41	Kygo		edm	228	68	69	-7
##	42	Taylor Swift		dance pop	171	68	77	-6
##	43	Lady Gaga		dance pop	216	39	57	-6
##	44	Khalid		pop	198	40	90	-9
##	45	ROSALÍA	r&b	en espanol	162	69	88	-4
##	46	Marshmello		brostep	182	62	66	-2
##	47	Nicky Jam		latin	202	75	67	-4
##	48	Marshmello		brostep	214	79	69	-3
##	49	The Chainsmokers		edm	218	70	59	-6
##	50	Ed Sheeran		pop	206	79	75	-6
##		Good mood	Duration	Popularity	Speechiness			
##	1	75	8	79	3			
##	2	61	8	92	9			
##	3	70	16	85	46			
##	4	55	8	86	19			
##	5	18	11	94	7			
##	6	84	9	84	4			
##	7	23	7	92	29			
##	8	35	8	90	9			
##	9	64	11	87	10			
##	10	56	10	95	38			
##	11	24	24	93	31			
##	12	38	15	86	7			
##	13	45	11	88	3			
##	14	76	6	87	20			
##	15	10	12	92	5			
##	16	14	16	82	15			
##	17	43	11	90	32			
##	18	91	7	91	5			
##	19	50	14	88	8			
##	20	41	12	91	11			
##	21	63	7	91	3			
##	22	48	12	91	8			
##	23	58	9	83	7			
##	24	80	44	91	4			
##	25	68	12	89	30			
##	26	87	13	70	6			
##	27	54	18	83	10			
##	28	40	19	89	3			
##	29	66	6	91	6			
##	30	94	36	89	25			
##	31	36	10	84	4			
##	32	33	9	89	33			
##	33	68	9	89	28			
##	34	18	13	89	15			
##	35	39	32	89	6			

## 36	32	14	89	6
## 37	62	11	91	34
## 38	91	36	87	5
## 39	95	11	80	6
## 40	68	6	78	7
## 41	40	10	88	3
## 42	73	7	90	5
## 43	32	23	87	3
## 44	35	6	84	13
## 45	75	5	88	12
## 46	44	58	88	5
## 47	80	8	88	6
## 48	67	17	88	5
## 49	50	41	88	3
## 50	61	7	82	12

7/ fonction "select"

La fonction select permet de sélectionner une colonne de la table, la commande pour cela est semblable en certains points à une commande SQL.

```
dabr::select(conn, "SELECT BPM FROM top50")
```

```
## Executing:
```

```
## SELECT BPM FROM top50
```

```
##
```

```
## Results: 50 records were found.
```

```
##      BPM
```

```
## 1  191
```

```
## 2  302
```

```
## 3  186
```

```
## 4  198
```

```
## 5  175
```

```
## 6  220
```

```
## 7  131
```

```
## 8  202
```

```
## 9  157
```

```
## 10 194
```

```
## 11 251
```

```
## 12 185
```

```
## 13 182
```

```
## 14 288
```

```
## 15 205
```

```
## 16 261
```

```
## 17 243
```

```
## 18 158
```

```
## 19 161
```

```
## 20 173
```

```
## 21 153
```

```
## 22 115
## 23 309
## 24 266
## 25 195
## 26 191
## 27 210
## 28 213
## 29 193
## 30 222
## 31 210
## 32 179
## 33 176
## 34 200
## 35 224
## 36 164
## 37 226
## 38 162
## 39 181
## 40 199
## 41 228
## 42 171
## 43 216
## 44 198
## 45 162
## 46 182
## 47 202
## 48 214
## 49 218
## 50 206
```

8/ fonction “update”

```
update <- function(conn, ...) {
  UseMethod("update", conn)
}
#dabr::update(conn, query, quiet = FALSE)
```

9/ La fonction “get_attr”

Cette fonction permet d’avoir la liste des attributs de la table que l’on utilise.

```
get_attr <- function(conn, ...) {
  UseMethod("get_attr", conn)
}
dabr::get_attr(conn, "top50")

## [1] "id"          "Track Name"  "Artist Name" "Genre"       "BPM"
## [6] "Energy %"    "Danceability" "Loudness dB"  "Good mood"   "Duration"
## [11] "Popularity"  "Speechiness"
```

10/ La fonction “quote”

Cette fonction permet tout simplement d’ajouter une citation sous forme de string.

```
quote <- function(conn, ...) {  
  UseMethod("quote", conn)  
}  
  
dabr::quote("Cette Database regroupe les 50 musiques les plus écoutées sur sp  
otify en 2019")  
  
## [1] "'Cette Database regroupe les 50 musiques les plus écoutées sur spotif  
y en 2019'"
```