`dbplyr` Package Tutorial

Jake Eisaguirre

Table of contents

`dbplyr` Package Information 2
Packages 2
Database Connection 2
`survey_data` Schema 3
Interacting with `dbplyr` 4

dbplyr Package Information

The dbplyr package is a user friendly and versatile package that can be used to interact with our ribbitr database. This package is a great tool for interacting with databases using tidyverse/dplyr syntax. dbplyr is the database back-end for the dplyr package which includes many of the user friend functions like filter(), select(), mutate(), and case_when(). The dbplyr package allows you to use remote database tables as if they are in-memory data frames by automatically converting dplyr code into SQL.

Packages

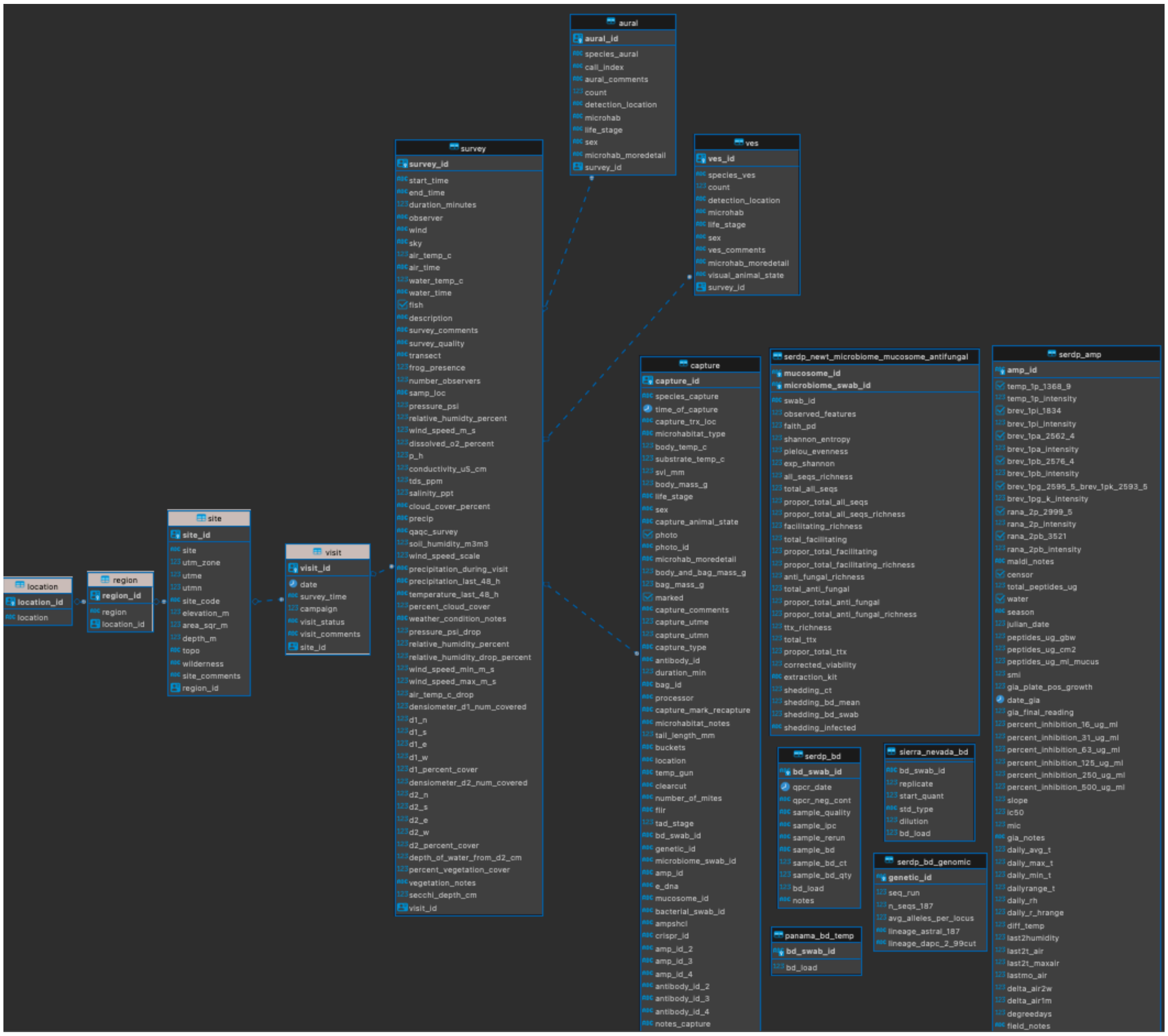
```r
if (!require(librarian)){
  install.packages("librarian")
  library(librarian)
}

# librarian downloads, if not already downloaded, and reads in needed packages
librarian::shelf(tidyverse, DBI, RPostgres, dbplyr, kableExtra)
```

Database Connection

Please see Data Base Connection Tutorial or reach out to me for more guidance on connecting to our ribbitr database.

```r
tryCatch({
    print("Connecting to Database…")
    connection <- dbConnect(drv = dbDriver("Postgres"),
                            dbname = Sys.getenv("aws_dbname"),
                            host = Sys.getenv("aws_host"),
                            port = Sys.getenv("aws_port"),
                            user = Sys.getenv("aws_user"),
                            password = Sys.getenv("aws_password"),
                            timezone=NULL)
    print("Database Connected!")
    },
    error=function(cond) {
            print("Unable to connect to Database.")})
# set search path for 'survey_data' schema
dbExecute(connection, "set search_path = 'survey_data'")
```

survey_data Schema



Now if you remember from previous database discussions, we know that most of the tables can be joined onto one another through what is called a `primary key` and `foreign key`. For instance, if we want to join the `location` table onto the `region` table, we would join the `location` tables `primary key`, which is called `location_id`, onto the `region` tables `foreign key`, which is also called `location_id`. In R, that would look something like this, `left_join(location, region, by = c("location_id")`.

So now by utilizing the `dbplyr` package, we can apply our understanding of data wrangling within R and convert those strings of `tidyverse/dplyr` commands into `SQL`. Once converted to a `SQL` command we can then send that query to the database.

Interacting with `dbplyr`

Using the `tbl()` functions from the `dbplyr` package stores a database version of the table in your local environment. You can then operate on those tables as if they are normal data frames in your RStudio environment.

Just like with all the `DBI` database functions, we must specify our `connection` to the database and then the table we are interested in storing. When specifying a table using the `dbplyr` package, you can think of it as always being in this format `tbl(connection, "insert_table_name")`.

If you want to see the `SQL` query used to retrieve that table you can use `show_query()`.

Now if you want to execute the query and retrieve the data from the database you would use `collect()`.

```
# Storing a database version table in memory of the `location` table
location_table <- tbl(connection, "location")


# Display SQL query
tbl(connection, "location") %>%
  show_query()
```

```
<SQL>
SELECT *
FROM "location"
```

```
# Retrieve data from the database
location_table <- tbl(connection, "location") %>%
  collect()
```

location	location_id
panama	78318db1-4920-4eb9-9f0f-c85a29950b77
brazil	db628122-7c2d-4401-a28e-f12f25b8266b
usa	e05b08d0-c2a6-4ba4-90a9-e12dbfafa63c

```
# Join `location` table onto `region` table by `location_id` and select columns
# of interest
loc_reg <- tbl(connection, "location") %>%
  left_join(tbl(connection, "region"), by = c("location_id")) %>%
  select(c(location, region)) %>%
  collect()
```

location	region
brazil	santa_virginia
brazil	boraceia
panama	fortuna
panama	santa_fe
panama	altos_de_campana
panama	chiriqui
panama	caribbean
panama	el_valle
panama	el_cope
panama	gamboa
usa	pennsylvania
usa	vermont
usa	new_mexico
usa	tennessee
usa	louisiana
usa	california

Now that we know the 3 basic functions, `tbl()`, `show_query()`, and `collect()`, from the `dbplyr` package we can try some more challenging data wrangling.

Columns of interest: `location`, `region`, `site`, `date`, `start_time`, `end_time`, `duration_minutes`, `species_captured`, `body_mass_g`, `svl_mm`, `life_stage`, and `sex`

```r
# Database version table in memory using `tidyverse`/`dplyr` language
db_data <- tbl(connection, "location") %>%
  left_join(tbl(connection, "region"), by = c("location_id")) %>%
  left_join(tbl(connection, "site"), by = c("region_id")) %>%
  left_join(tbl(connection, "visit"), by = c("site_id")) %>%
  left_join(tbl(connection, "survey"), by = c("visit_id")) %>%
  left_join(tbl(connection, "capture"), by = c("survey_id")) %>%
  select(c(location, region, site, date, start_time, end_time, duration_minutes,
           species_capture, body_mass_g, svl_mm, life_stage, sex))

# Retrieve data
clean_data <- db_data %>%
  collect()

# Show query
# in_memory_data %>%
#   show_query()

# Note: The method in how `dbplyr` creates the `SQL` query from the in memory data set
# is not the most efficient query. However, if you ran that query in `dbGetQuery` it would
# return the same results.
```

location	region	site	date	start_time	end_time	duration_minutes	species_capture	body_mass_g	svl_mm	life_stage	sex
brazil	santa_virginia	lago_sede_water	2020-12-13	NA	NA	NA	dendropsophus_minutus	NA	NA	NA	NA
brazil	santa_virginia	lago_sede_water	2020-12-13	NA	NA	NA	boana_bandeirantes	NA	NA	NA	NA
brazil	santa_virginia	lago_sede_water	2020-12-13	NA	NA	NA	dendropsophus_minutus	NA	NA	NA	NA
brazil	santa_virginia	lago_sede_water	2020-12-13	NA	NA	NA	boana_bandeirantes	NA	NA	NA	NA
brazil	santa_virginia	lago_sede_water	2020-12-13	NA	NA	NA	boana_bandeirantes	NA	NA	NA	NA
brazil	santa_virginia	lago_sede_water	2020-12-13	NA	NA	NA	boana_bandeirantes	NA	NA	NA	NA
panama	gamboa	gamboa	2016-02-18	19:00:00	NA	NA	craugastor_fitzingeri	NA	46.00	adult	unknown
brazil	santa_virginia	4_land	2020-12-05	17:05:00	18:20:00	75	brachycephalus_pitanga	NA	NA	NA	NA
panama	caribbean	miguel_de_la_borda	2014-07-17	09:36:00	12:25:00	169	incilius_coniferus	0.7	18.50	adult	unknown
panama	el_valle	mata_ahogado	2013-06-16	11:20:00	15:09:00	229	rhaebo_haematiticus	1.4	22.20	juvenile	unknown
panama	el_valle	mata_ahogado	2015-12-12	14:20:00	17:15:00	175	rhaebo_haematiticus	NA	NA	adult	unknown
panama	fortuna	alleman	2013-06-23	20:51:00	22:26:00	95	espadarana_prosoblepon	1.3	25.20	adult	male
panama	fortuna	alleman	2013-06-23	20:51:00	22:26:00	95	espadarana_prosoblepon	0.7	24.60	adult	male
brazil	santa_virginia	4_land	2020-12-05	NA	NA	NA	physalaemus_olfersii	NA	NA	NA	NA
brazil	santa_virginia	4_land	2020-12-05	17:05:00	18:20:00	75	brachycephalus_pitanga	NA	NA	NA	NA
brazil	santa_virginia	4_land	2020-12-05	17:05:00	18:20:00	75	brachycephalus_pitanga	NA	NA	NA	NA
panama	el_cope	rio_marta	2016-07-05	10:05:00	13:35:00	210	colostethus_panamensis	1.6	23.80	adult	male
brazil	santa_virginia	4_land	2020-12-05	NA	NA	NA	dendrophryniscus_haddadi	NA	NA	NA	NA
panama	el_cope	rio_marta	2018-11-28	09:00:00	10:25:00	85	colostethus_panamensis	NA	25.20	adult	female
brazil	santa_virginia	4_land	2020-12-05	17:05:00	18:20:00	75	physalaemus_olfersii	NA	NA	NA	NA
panama	caribbean	sargentita	2015-07-22	20:26:00	22:20:00	114	leptodactylus_savagei	NA	NA	adult	female
panama	el_cope	rio_tigrero	2018-11-19	10:53:00	14:00:00	187	colostethus_panamensis	0.7	27.90	adult	unknown
panama	el_cope	sophia_stream	2019-12-11	10:18:00	12:59:00	161	silverstoneia_flotator	0.4	17.78	adult	NA
panama	el_cope	sophia_stream	2022-07-28	14:01:00	15:51:00	110	unknown	NA	NA	tadpole	NA
brazil	santa_virginia	4_land	2020-12-05	NA	NA	NA	brachycephalus_pitanga	NA	NA	NA	NA

Now we can run the same query as above but incorporating more data wrangling on the database version of the tables. Lets say we are only interested in organisms greater then 32 mm svl, are heavier then 25 g, who are all adults, are from panama and the usa, and with a date range from 2015 to present. And for fun we also want to convert the svl from mm to cm.

```r
# In memory storage of data selection using `tidyverse`/`dplyr` language
db_data <- tbl(connection, "location") %>%
  left_join(tbl(connection, "region"), by = c("location_id")) %>%
  left_join(tbl(connection, "site"), by = c("region_id")) %>%
  left_join(tbl(connection, "visit"), by = c("site_id")) %>%
  left_join(tbl(connection, "survey"), by = c("visit_id")) %>%
  left_join(tbl(connection, "capture"), by = c("survey_id")) %>%
  select(c(location, region, site, date,
           species_capture, svl_mm, body_mass_g, life_stage, sex)) %>%
  filter(location %in% c("panama", "usa"),
         svl_mm > 32,
         body_mass_g > 25,
         life_stage == "adult",
         date > "2015-01-01") %>%
  rename(svl_cm = svl_mm) %>%
  mutate(svl_cm = svl_cm / 10)

# Retrieve data
clean_data <- db_data %>%
  collect()
```

location	region	site	date	species_capture	svl_cm	body_mass_g	life_stage	sex
panama	santa_fe	altos_de_piedra	2019-08-02	rhinella_marina	14.200	71.00	adult	unknown
panama	el_cope	guabal	2019-08-08	rhaebo_haematiticus	9.200	37.50	adult	unknown
panama	altos_de_campana	rabbit_stream	2015-06-25	rhaebo_haematiticus	7.420	29.60	adult	unknown
panama	altos_de_campana	rana_dorada	2016-12-12	rhaebo_haematiticus	6.400	25.35	adult	unknown
panama	el_cope	rio_marta	2015-06-22	rhinella_marina	8.160	48.00	adult	unknown
panama	el_cope	medina	2022-07-27	rhaebo_haematiticus	7.042	29.60	adult	unkonwn
panama	el_cope	rio_tigrero	2018-11-21	rhaebo_haematiticus	7.110	25.50	adult	unknown
panama	el_cope	sophia_stream	2019-08-07	unknown_species	7.400	32.10	adult	unknown
usa	pennsylvania	admin_pond	2022-05-19	rana_catesbeiana	6.610	31.70	adult	female
usa	pennsylvania	admin_pond	2022-05-19	rana_catesbeiana	7.100	31.50	adult	female
usa	pennsylvania	tuttle_pond	2022-06-14	rana_clamitans	6.470	26.70	adult	male
usa	pennsylvania	tuttle_pond	2022-06-14	rana_catesbeiana	13.110	200.00	adult	female
usa	pennsylvania	rv_pond	2022-06-07	rana_catesbeiana	11.950	200.00	adult	female
usa	pennsylvania	tuttle_pond	2022-06-14	rana_catesbeiana	7.610	46.50	adult	female
usa	pennsylvania	tuttle_pond	2022-06-15	rana_catesbeiana	6.280	28.30	adult	female
usa	pennsylvania	rv_pond	2022-06-08	rana_catesbeiana	8.830	68.00	adult	female
usa	pennsylvania	tuttle_pond	2022-06-15	rana_catesbeiana	6.800	38.80	adult	female
usa	pennsylvania	rv_pond	2022-06-08	rana_clamitans	7.060	42.20	adult	female
usa	pennsylvania	rv_pond	2022-06-08	rana_catesbeiana	11.950	200.00	adult	female
usa	pennsylvania	tuttle_pond	2022-06-14	rana_catesbeiana	7.900	52.10	adult	female