Access Wideband Audiology Immitance SQL database using R and dplyr (Voss PI)

Nicholas Horton (nhorton@amherst.edu)

2025-05-20

Introduction

This document is intended to describe how to access data from a MySQL database using R. It utilizes a database of wideband acoustic immitance variables from humans with normal hearing (see https://projectreporter.nih.gov/project_info_description.cfm?aid=8769352&icde=30039221&ddparam=&c for more details).

A relevant paper on the topic of data management and databases in R can be found at http://chance.amstat.org/2015/04/setting-the-stage.

See also

- https://nicholasjhorton.github.io/K12-Data-Tools/icerm.html.
- https://hardin47.netlify.app/courses/sds261-sql
- https://beanumber.github.io/tidy-databases/
- https://mdsr-book.github.io/mdsr3e/15-sqlI.html

Accessing data from a database using SQL commands

First I will demonstrate how to access data using SQL (structured query language) commands and the `dbGetQuery()` function. We begin by setting up a connection to the database.

```
library(mosaic)
library(tidyverse)
db <- DBI::dbConnect(
  RMySQL::MySQL(),
  dbname = "wai",
  host = "scidb.smith.edu",
  user = "waiuser",
```

```
  password = "smith_waiDB"
)
```

Next a series of SQL queries can be sent to the database. These return R dataframes.

```
dbGetQuery(db, "SHOW TABLES")
```

```
         Tables_in_wai
1             Codebook
2         Measurements
3 Measurements_pre2020
4              PI_Info
5          PI_Info_OLD
6             Subjects
7     Subjects_pre2020
```

```
dbGetQuery(db, "EXPLAIN PI_Info")
```

```
                Field        Type Null Key Default Extra
1          Identifier varchar(50)   NO PRI    <NA>
2                Year         int   NO        <NA>
3             Authors        text   NO        <NA>
4    AuthorsShortList        text   NO        <NA>
5               Title        text   NO        <NA>
6             Journal        text   NO        <NA>
7                 URL        text   NO        <NA>
8            Abstract        text   NO        <NA>
9   DataSubmitterName        text   NO        <NA>
10 DataSubmitterEmail        text   NO        <NA>
11      DateSubmitted        text   NO        <NA>
12           PI_Notes        text   NO        <NA>
```

```
dbGetQuery(db, "EXPLAIN Subjects")
```

```
                          Field        Type Null Key Default Extra
1                    Identifier varchar(50)   NO PRI    <NA>
2                 SubjectNumber         int   NO PRI    <NA>
3                  SessionTotal         int   NO        <NA>
4             AgeFirstMeasurement       float  YES        <NA>
```

```
5     AgeCategoryFirstMeasurement varchar(50)  YES        <NA>
6                             Sex varchar(50)   NO        <NA>
7                            Race varchar(50)   NO        <NA>
8                       Ethnicity varchar(50)   NO        <NA>
9   LeftEarStatusFirstMeasurement varchar(50)   NO        <NA>
10 RightEarStatusFirstMeasurement varchar(50)   NO        <NA>
11                   SubjectNotes        text  YES        <NA>
```

```
dbGetQuery(db, "EXPLAIN Measurements")
```

```
            Field        Type Null Key Default Extra
1      Identifier varchar(50)   NO PRI    <NA>
2   SubjectNumber         int   NO PRI    <NA>
3         Session         int   NO PRI    <NA>
4             Ear varchar(50)   NO PRI
5      Instrument varchar(50)   NO PRI
6             Age       float  YES        <NA>
7     AgeCategory varchar(50)  YES        <NA>
8       EarStatus varchar(50)  YES        <NA>
9             TPP       float  YES        <NA>
10      AreaCanal       float  YES        <NA>
11  PressureCanal       float   NO PRI       0
12 SweepDirection varchar(50)   NO PRI
13      Frequency       float   NO PRI       0
14     Absorbance       float  YES        <NA>
15           Zmag       float  YES        <NA>
16           Zang       float  YES        <NA>
```

```
first_ten <- dbGetQuery(db, "SELECT * from Measurements LIMIT 10")
first_ten
```

```
   Identifier SubjectNumber Session  Ear Instrument Age AgeCategory EarStatus
1   Abur_2014             1       1 Left     HearID  20       Adult    Normal
2   Abur_2014             1       1 Left     HearID  20       Adult    Normal
3   Abur_2014             1       1 Left     HearID  20       Adult    Normal
4   Abur_2014             1       1 Left     HearID  20       Adult    Normal
5   Abur_2014             1       1 Left     HearID  20       Adult    Normal
6   Abur_2014             1       1 Left     HearID  20       Adult    Normal
7   Abur_2014             1       1 Left     HearID  20       Adult    Normal
8   Abur_2014             1       1 Left     HearID  20       Adult    Normal
9   Abur_2014             1       1 Left     HearID  20       Adult    Normal
```

```
10  Abur_2014             1       1 Left     HearID  20       Adult    Normal
   TPP AreaCanal PressureCanal SweepDirection Frequency Absorbance      Zmag
1   -5  4.42e-05             0        Ambient   210.938  0.0333379 113780000
2   -5  4.42e-05             0        Ambient   234.375  0.0315705 103585000
3   -5  4.42e-05             0        Ambient   257.812  0.0405751  92951700
4   -5  4.42e-05             0        Ambient   281.250  0.0438399  86058000
5   -5  4.42e-05             0        Ambient   304.688  0.0486400  79492800
6   -5  4.42e-05             0        Ambient   328.125  0.0527801  73326200
7   -5  4.42e-05             0        Ambient   351.562  0.0583192  68793600
8   -5  4.42e-05             0        Ambient   375.000  0.0638881  64088600
9   -5  4.42e-05             0        Ambient   398.438  0.0687025  60200600
10  -5  4.42e-05             0        Ambient   421.875  0.0833181  56990900
        Zang
1  -0.233504
2  -0.235778
3  -0.233482
4  -0.233421
5  -0.232931
6  -0.232837
7  -0.232115
8  -0.231642
9  -0.231356
10 -0.228356
```

Accessing a database using dplyr commands

Alternatively, a connection can be made to the server by creating a series of dplyr table objects.

```
Measurements <- tbl(db, "Measurements")
PI_Info <- tbl(db, "PI_Info")
Subjects <- tbl(db, "Subjects")
```

Let's explore the `PI_Info` table.

```
PI_Info |>
  collect() |>
  summarise(total = n())
```

```
# A tibble: 1 x 1
```

```
  total
  <int>
1    43
```

```
PI_Info |>
  collect() |>
  glimpse()   # collect() is a bad idea when dealing with large tables!
```

```
Rows: 43
Columns: 12
$ Identifier         <chr> "Abur_2014", "Aithal_2013", "Aithal_2014", "Aithal_~
$ Year               <int> 2014, 2013, 2014, 2014, 2015, 2017, 2017, 2019, 201~
$ Authors            <chr> "Defne Abur, Nicholas J. Horton, and Susan E. Voss"~
$ AuthorsShortList   <chr> "Abur et al.", "Aithal et al.", "Aithal et al.", "A~
$ Title              <chr> "Instrasubject variability in power reflectance", "~
$ Journal            <chr> "J Am Acad Audiol", "Int. J. Pediatr. Otorhinolaryn~
$ URL                <chr> "https://www.ncbi.nlm.nih.gov/pubmed/25257718", "ht~
$ Abstract           <chr> "\"<p> <strong> Background: </strong> Power reflect~
$ DataSubmitterName  <chr> "Susan Voss", "Sreedevi Aithal", "Sreedevi Aithal",~
$ DataSubmitterEmail <chr> "svoss@smith.edu", "Sreedevi.aithal@health.qld.gov.~
$ DateSubmitted      <chr> "24-Aug-2016", "2-Jan-2023", "2-Jan-2023", "26-Feb-~
$ PI_Notes           <chr> "Measurements made on 7 subjects across multiple se~
```

Let's explore the `Subjects` table.

```
Subjects |>
  collect()  # be careful with collect() with large tables!
```

```
# A tibble: 10,108 x 11
   Identifier  SubjectNumber SessionTotal AgeFirstMeasurement
   <chr>               <int>        <int>               <dbl>
 1 Abur_2014               1            7              20
 2 Abur_2014               3            8              19
 3 Abur_2014               4            7              21
 4 Abur_2014               6            8              21
 5 Abur_2014               7            5              20
 6 Abur_2014               8            5              19
 7 Abur_2014              10            5              19
 8 Aithal_2013             1            1              NA
 9 Aithal_2013             2            1               0.00744
10 Aithal_2013             3            1              NA
```

```
# i 10,098 more rows
# i 7 more variables: AgeCategoryFirstMeasurement <chr>, Sex <chr>, Race <chr>,
#   Ethnicity <chr>, LeftEarStatusFirstMeasurement <chr>,
#   RightEarStatusFirstMeasurement <chr>, SubjectNotes <chr>
```

Let's explore the `Measurements` table.

```
Measurements |>
  summarise(total = n())
```

```
# Source:   SQL [?? x 1]
# Database: mysql 8.0.41-0ubuntu0.20.04.1 [@scidb.smith.edu:/wai]
    total
    <dbl>
1 4981558
```

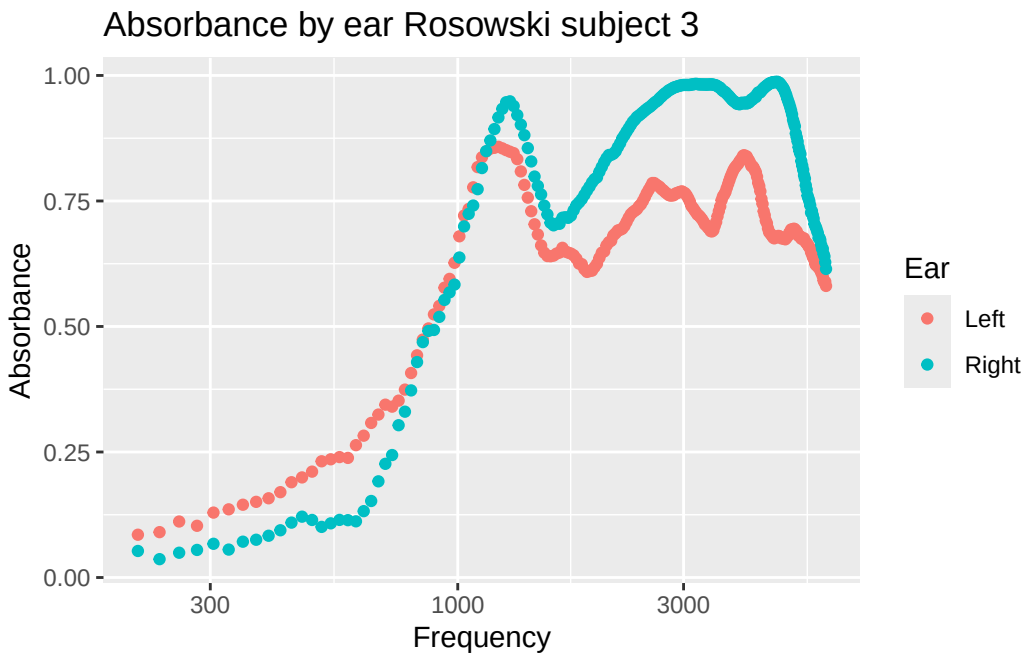
Let's download the data from a given subject from the Rosowski_2012 study

```
onesubj <- Measurements |>
  filter(Identifier == "Rosowski_2012", SubjectNumber == 3) |>
  collect() |>
  mutate(SessionNum = as.factor(Session))
head(onesubj)
```

```
# A tibble: 6 x 17
  Identifier  SubjectNumber Session Ear   Instrument   Age AgeCategory EarStatus
  <chr>               <int>   <int> <chr> <chr>      <dbl> <chr>       <chr>
1 Rosowski_2~             3       1 Left  HearID        30 Adult       Normal
2 Rosowski_2~             3       1 Left  HearID        30 Adult       Normal
3 Rosowski_2~             3       1 Left  HearID        30 Adult       Normal
4 Rosowski_2~             3       1 Left  HearID        30 Adult       Normal
5 Rosowski_2~             3       1 Left  HearID        30 Adult       Normal
6 Rosowski_2~             3       1 Left  HearID        30 Adult       Normal
# i 9 more variables: TPP <dbl>, AreaCanal <dbl>, PressureCanal <dbl>,
#   SweepDirection <chr>, Frequency <dbl>, Absorbance <dbl>, Zmag <dbl>,
#   Zang <dbl>, SessionNum <fct>
```

Finally we can plot the absorbance as a function of frequency, stratified by ear.

```
ggplot(
  data = onesubj,
  aes(x = Frequency, y = Absorbance)) +
    geom_point() +
    aes(colour = Ear) +
    scale_x_log10() +
    labs(title = "Absorbance by ear Rosowski subject 3")
```



CHALLENGE #1: In groups of two: write SQL code to calculate which study has the largest number of measurements (rows)? Which study has the largest number of subjects?

CHALLENGE #2: In groups of two: write SQL code to find how many frequencies are measured in the Voss_1994 study as compared to the Voss_2010 study.

```
dbDisconnect(db) # please be polite and close the connection
```

```
[1] TRUE
```