Exercise 1: Connecting to the database

Load the necessary packages and open a connection named `lahman_db_2019` to the Lahman 2019 season database housed in the `sqlite` file in the data folder.

Solutions:

```
library(RSQLite)
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

```
library(DBI)
lahman_db_2019 <- dbConnect(SQLite(), "./data/2019_lahmansbaseballdb.sqlite")
dbExecute(lahman_db_2019, "pragma foreign_keys = on")
```

```
## [1] 0
```

Exercise 2: Getting to know your database

(a) List all relations in the database

Solutions:

dbListFields(lahman_db_2019)

(b) Consider the relation named `Batting`/. Save it as a data frame in your `R` session, called `batting_2019`. Check that `batting_2019` is indeed a data frame. What is its dimension?

Solutions:

```
batting <- dbSendQuery(lahman_db_2019,
                       "SELECT * from batting")
batting_2019 <- dbFetch(batting)
print(sum(batting_2019$HR))
```

```
## [1] 307761
```

(c) Remove `eval=FALSE` from the `R` code chunks below and run the code chunks. Then, after each SQL query (each call to `dbGetQuery()`) has executed, explain in words what is being extracted.

(i)

```
dbGetQuery(lahman_db_2019,
           "SELECT playerID, yearID, AB, H, HR
           FROM Batting
           ORDER BY yearID
           LIMIT 10"
)
```

(ii)

```
dbGetQuery(lahman_db_2019,
           "SELECT playerID, yearID, AB, H, HR
           FROM Batting
           ORDER BY HR DESC
           LIMIT 10"
)
```

```
##     playerID yearID  AB   H HR
## 1  bondsba01   2001 476 156 73
## 2  mcgwima01   1998 509 152 70
## 3   sosasa01   1998 643 198 66
## 4  mcgwima01   1999 521 145 65
## 5   sosasa01   2001 577 189 64
## 6   sosasa01   1999 625 180 63
## 7  marisro01   1961 590 159 61
## 8   ruthba01   1927 540 192 60
## 9   ruthba01   1921 540 204 59
## 10 stantmi03   2017 597 168 59
```

(iii)

```
dbGetQuery(lahman_db_2019,
           "SELECT playerID, yearID, AB, H, HR
           FROM Batting
           WHERE HR > 50
           ORDER BY HR DESC"
)
```

```
##     playerID yearID  AB   H HR
## 1  bondsba01   2001 476 156 73
## 2  mcgwima01   1998 509 152 70
## 3   sosasa01   1998 643 198 66
## 4  mcgwima01   1999 521 145 65
## 5   sosasa01   2001 577 189 64
## 6   sosasa01   1999 625 180 63
## 7  marisro01   1961 590 159 61
## 8   ruthba01   1927 540 192 60
## 9   ruthba01   1921 540 204 59
## 10 stantmi03   2017 597 168 59
## 11  foxxji01   1932 585 213 58
## 12 greenha01   1938 556 175 58
## 13 howarry01   2006 581 182 58
## 14 gonzalu01   2001 609 198 57
## 15 rodrial01   2002 624 187 57
```

Exercise 3: SQL computations

(a) As before, remove `eval=FALSE` from the following `R` code chunks. Then, after each SQL query, explain in words what is being extracted.

(i)

```
dbGetQuery(lahman_db_2019,
           "SELECT AVG(HR)
           FROM BATTING"
)
```

```
##    AVG(HR)
## 1 2.864785
```

Exercise 4: Some more practice with SQL computations

(a) Use a SQL query on the `Batting` relation to calculate each player's average number of hits (`H`) over the seasons they played, and display the players with the 10 highest hit averages, along with their hit averages. **Hint**: `AVG()`, `GROUP BY`, `ORDER BY`.

Solutions:

```
dbGetQuery(lahman_db_2019,
"SELECT playerid, MAX(yearID), AVG(H) as average
FROM Batting
GROUP BY playerID
ORDER BY AVG(H) DESC
LIMIT 10")
```

```
##     playerID MAX(yearID)  average
## 1  puckeki01        1995 192.0000
## 2  burkeje01        1905 178.1250
## 3  sislege01        1930 175.7500
## 4   cobbty01        1928 174.5417
## 5  altuvjo01        2019 174.2222
## 6  jeterde01        2014 173.2500
## 7  abreujo02        2019 173.0000
## 8  ashburi01        1962 171.6000
## 9   canoro01        2019 171.3333
## 10 dimagjo01        1951 170.3077
```