# Data607-Assignment2

*Scott Reed*

*9/5/2019*

## Database and R

---

*Note:* Please configure to a blank postgres db (use: createdb DBNAME) with the proper connection strings

---

### Postgress connection

## Data table setup

We use three data tables one movies, one for raters and then a join table for ratings.

```
dbExecute(conn, "CREATE TABLE IF NOT EXISTS movies(
    movie_id  INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    movie_name VARCHAR NOT NULL
);")
```

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

```
dbExecute(conn, "CREATE TABLE IF NOT EXISTS raters(
    rater_id  INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    rater_name VARCHAR NOT NULL
);")
```

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

```
dbExecute(conn, "CREATE TABLE IF NOT EXISTS movieratings(
    rating_id  INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    rater_id INT NOT NULL REFERENCES raters(rater_id),
    movie_id INT NOT NULL REFERENCES movies(movie_id),
    rating decimal
);")
```

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

### Table setup

### Movies

| field | type |
|---|---|
| movie_id | int (auto) |
| movie_name | varchar |

**Raters**

| field | type |
|---|---|
| rater_id | int (auto) |
| rater_name | varchar |

**MovieRatings**

| field | type |
|---|---|
| rating_id | int (auto) |
| rater_id | int (fk) |
| movie_id | int (fk) |
| rating | decimal |

**get some names**

We shall grab some name data from NYC and sample a number of them

```
names <- read.csv("https://data.cityofnewyork.us/api/views/25th-nujf/rows.csv?accessType=DOWNLOAD")
someNames <- sample(as.character(names[,4]),50)
head(someNames)
```

```
## [1] "LINDSAY" "ELIJAH"  "Quinn"   "Julien"  "Elliot"  "Ella"
```

We then write them to the database, and read it back for the IDs.

```
dbBegin(conn)
```

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

```
tblRaters <- as.data.frame(someNames)
names(tblRaters) <- c("rater_name")
dbExecute(conn,sqlAppendTable(conn, "raters", tblRaters,row.names = FALSE ))
```

```
## [1] 50
```

```
dbCommit(conn)
```

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

```
tblRaters = dbReadTable(conn,"raters")
head(tblRaters)
```

```
##   rater_id rater_name
## 1        1    LINDSAY
## 2        2     ELIJAH
## 3        3      Quinn
## 4        4     Julien
## 5        5     Elliot
## 6        6       Ella
```

### Get some movies

We get some movies sample them and load them into a database table much like names

```
load(url("https://stat.duke.edu/~mc301/data/movies.Rdata"))
someMovies <- sample(as.character(movies$title),6)
head(someMovies)
```

```
## [1] "The Babysitter"             "Aladdin"
## [3] "The Astronaut Farmer"       "Jennifer 8"
## [5] "U-Turn"                     "Ice Age: Continental Drift"
```

```
dbBegin(conn)
```

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

```
tblMovies <- as.data.frame(someMovies)
names(tblMovies) <- c("movie_name")
dbExecute(conn,sqlAppendTable(conn, "movies", tblMovies,row.names = FALSE ))
```

```
## [1] 6
```

```
dbCommit(conn)
```

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

```
tblMovies = dbReadTable(conn,"movies")
kable(tblMovies)
```

| movie_id | movie_name |
|---|---|
| 1 | The Babysitter |
| 2 | Aladdin |
| 3 | The Astronaut Farmer |
| 4 | Jennifer 8 |
| 5 | U-Turn |
| 6 | Ice Age: Continental Drift |

### Add some ratings

To generate some ratings we first cross apply movie_id and rater_id and then provide ratings up to 6. We then take any over 5 (our upper bound) to be nulls.

```
tblRatings<-expand.grid(tblMovies$movie_id,tblRaters$rater_id)
names(tblRatings) <- c("movie_id", "rater_id")
tblRatings$rating <-  sample(6, size = nrow(tblRatings), replace = TRUE)
tblRatings[which(tblRatings["rating"] > 5),]$rating <- NA
tail(tblRatings)
```

```
##     movie_id rater_id rating
## 295        1       50      5
## 296        2       50      3
## 297        3       50      2
## 298        4       50      3
## 299        5       50     NA
## 300        6       50      4
```

We then store and fetch our ratings

```
dbBegin(conn)
```

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

```
dbExecute(conn,sqlAppendTable(conn, "movieratings", tblRatings,row.names = FALSE ))
```

```
## [1] 300
```

```
dbCommit(conn)
```

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

```
tblRatings = dbReadTable(conn,"movieratings")
kable(head(tblRatings))
```

| rating_id | rater_id | movie_id | rating |
|---|---|---|---|
| 1 | 1 | 1 | 5 |
| 2 | 1 | 2 | 2 |
| 3 | 1 | 3 | 1 |
| 4 | 1 | 4 | 5 |
| 5 | 1 | 5 | 5 |
| 6 | 1 | 6 | 2 |

### Joining and getting summary data

If we want we can get some summary data for our ratings but first we must join them.

```
tblAvg <- dbGetQuery(conn,"SELECT movie_name, avg(rating) as avgrating
   from movieratings INNER JOIN movies ON movieratings.movie_id = movies.movie_id  INNER JOIN raters ON
kable(head(tblAvg))
```

| movie_name | avgrating |
|---|---|
| Aladdin | 3.476190 |
| Ice Age: Continental Drift | 3.066667 |
| Jennifer 8 | 3.023256 |
| The Astronaut Farmer | 2.813954 |
| The Babysitter | 3.214286 |
| U-Turn | 2.951220 |