STAT 33B Workbook 13

Yuanrui Zhu (3034615728)

Dec 3, 2020

This workbook is due **Dec 3, 2020** by 11:59pm PT.

The workbook is organized into sections that correspond to the lecture videos for the week. Watch a video, then do the corresponding exercises *before* moving on to the next video.

Workbooks are graded for completeness, so as long as you make a clear effort to solve each problem, you'll get full credit. That said, make sure you understand the concepts here, because they're likely to reappear in homeworks, quizzes, and later lectures.

As you work, write your answers in this notebook. Answer questions with complete sentences, and put code in code chunks. You can make as many new code chunks as you like.

In the notebook, you can run the line of code where the cursor is by pressing `Ctrl` + `Enter` on Windows or `Cmd` + `Enter` on Mac OS X. You can run an entire code chunk by clicking on the green arrow in the upper right corner of the code chunk.

Please do not delete the exercises already in this notebook, because it may interfere with our grading tools.

You need to submit your work in two places:

- Submit this Rmd file with your edits on bCourses.
- Knit and submit the generated PDF file on Gradescope.

If you have any last-minute trouble knitting, **DON'T PANIC**. Submit your Rmd file on time and follow up in office hours or on Piazza to sort out the PDF.

Relational Data

Watch the "Relational Data" lecture video.

No exercises for this section.

Joins (with dplyr)

Watch the "Joins (with dplyr)" lecture video.

In this workbook, you'll use three tables from the Internet Movie Database (IMDB) to practice joining data frames and taking subsets.

The following is a description of the three tables and their columns.

- Titles (`titles10s.rds`), where each row is one movie. Columns:

    - `tconst` - alphanumeric unique identifier of the title
    - `titleType` - the type of the title (movie or tvMovie)
    - `primaryTitle` - title used by the filmmakers at the point of release
    - `originalTitle` - original title, in the original language
    - `startYear` - release year of the title
    - `runtimeMinutes` - primary runtime of the title, in minutes

- Cast (`cast10s.rds`), where each row is one cast member from a movie. Columns:
    - `tconst` - alphanumeric unique identifier of the title
    - `ordering` - a number to uniquely identify rows for a given tconst
    - `nconst` - alphanumeric unique identifier of the name/person
    - `category` - the category of job that person was in
    - `job` - the specific job title if applicable, else `NA`
- People (`people10s.rds`), where each row is one person. Columns:
    - `nconst` - alphanumeric unique identifier of the name/person
    - `primaryName` - name by which the person is most often credited

The tables are a subset of IMDB’s larger collection of data, which is available at https://www.imdb.com/interfaces/.

Exercise 1

Find the names of the three people that had the most roles in movies in the data set.

Hint 1: Think about what the rows in each table represent. Is there a table where rows represent appearances/roles in movies?

Hint 2: A join is only needed here to get the names of the people.

YOUR ANSWER GOES HERE:

```
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
cast = readRDS("cast10s.rds")
people = readRDS("people10s.rds")
titles = readRDS("titles10s.rds")
```

```
# keep all the people with nconst and primary name
join1 = inner_join(cast, people, by = ’nconst’)
truncated1 = join1[, c("category", "primaryName")]
aggregated1 = aggregate(category ~ primaryName, truncated1, length)
head(aggregated1[order(aggregated1$category, decreasing = T), ])
```

```
##                 primaryName category
## 184163       Kevin MacLeod      281
## 99990         Eric Roberts      145
## 345955 William Shakespeare      137
## 44118         Brahmanandam      117
## 127336          Harvey Kahn      113
## 199365      Liverpool F.C.       93
```

From the table, we can see that Kevin Macleod, Eric Roberts and William Shakespeare are the top three characters who has the most roles in the movie.

Exercise 2

Compute a data frame that contains the `nconst` IDs, names, and roles of the primary cast from the 2018 movie “Black Panther”.

Hint 1: A common strategy for relational data is to reduce the size of a table or get specific rows by taking a subset, and then join that table with another table. Start by taking a subset of the Titles table to find the `tconst` ID for Black Panther.

Hint 2: When you have more than two tables, it is sometimes necessary to use more than one join.

YOUR ANSWER GOES HERE:

```
black1 = titles[which(titles$primaryTitle == "Black Panther"), ]
black2 = inner_join(cast, black1, by = ’tconst’)
black3 = inner_join(black2, people, by = ’nconst’)
black3[, c(’nconst’, ’primaryName’, ’category’)]
```

```
## # A tibble: 10 x 3
##    nconst    primaryName       category
##    <chr>     <chr>             <fct>
##  1 nm3234869 Ludwig Göransson  composer
##  2 nm1569276 Chadwick Boseman  actor
##  3 nm0430107 Michael B. Jordan actor
##  4 nm2143282 Lupita Nyong’o    actress
##  5 nm1775091 Danai Gurira      actress
##  6 nm3363032 Ryan Coogler      director
##  7 nm1963288 Joe Robert Cole   writer
##  8 nm0498278 Stan Lee          writer
##  9 nm0456158 Jack Kirby        writer
## 10 nm0270559 Kevin Feige       producer
```

Exercise 3

Compute a data frame that contains the names of the primary **actors and actresses** for all Harry Potter movies in the data set.

Hint: The `startsWith` function (or stringr) is helpful for identifying Harry Potter movies.

YOUR ANSWER GOES HERE:

```
harry1 = titles[which(startsWith(titles$primaryTitle, "Harry Potter")), ]
harry2 = inner_join(harry1, cast, by = 'tconst')
harry3 = inner_join(harry2, people, by = 'nconst')
harry4 = harry3[which(harry3$category == "actor" | harry3$category == "actress"), ]
harry4
```

```
## # A tibble: 8 x 11
##   tconst titleType primaryTitle originalTitle startYear runtimeMinutes ordering
##   <chr>  <fct>     <chr>        <chr>             <dbl>          <dbl>    <int>
## 1 tt092~ movie     Harry Potte~ Harry Potter~      2010            146        1
## 2 tt092~ movie     Harry Potte~ Harry Potter~      2010            146        2
## 3 tt092~ movie     Harry Potte~ Harry Potter~      2010            146        3
## 4 tt092~ movie     Harry Potte~ Harry Potter~      2010            146        4
## 5 tt120~ movie     Harry Potte~ Harry Potter~      2011            130        1
## 6 tt120~ movie     Harry Potte~ Harry Potter~      2011            130        2
## 7 tt120~ movie     Harry Potte~ Harry Potter~      2011            130        3
## 8 tt120~ movie     Harry Potte~ Harry Potter~      2011            130        4
## # ... with 4 more variables: nconst <chr>, category <fct>, job <chr>,
## #   primaryName <chr>
```

```
unique(harry4['primaryName'])
```

```
## # A tibble: 5 x 1
##   primaryName
##   <chr>
## 1 Daniel Radcliffe
## 2 Emma Watson
## 3 Rupert Grint
## 4 Bill Nighy
## 5 Michael Gambon
```

STAT 33 Wrap-up

Watch the “STAT 33 Wrap-up” lecture video.

Please fill in the teaching evaluations for this class!