Assignment 2

Yan Zichu, 300476924

30 July 2020

Q1. (1 Mark) Write R code to create and connect to a new SQLite database with storage file cricket.sqlite

ANS:

```
library(DBI)
library(RSQLite)
test_conn <- dbConnect(RSQLite::SQLite(), "cricket.sqlite")
```

Q2. (2 Marks) Write R code to read the seven data files into R. Combine the three outcomes data files (T20, ODI and Test) into a single R object, and report the numbers of rows it has.

ANS:

```
grounds <- read.csv("grounds.csv", stringsAsFactors=FALSE)
innings <- read.csv("innings.csv", stringsAsFactors=FALSE)
players <- read.csv("players.csv", stringsAsFactors=FALSE)
teams <- read.csv("teams.csv", stringsAsFactors=FALSE)
wcricketODI <- read.csv("wcricket-ODI.csv", stringsAsFactors=FALSE)
wcricketT20 <- read.csv("wcricket-T20.csv", stringsAsFactors=FALSE)
wcricketTest <- read.csv("wcricket-Test.csv", stringsAsFactors=FALSE)

filenames <- list.files(path="/Users/Patrick/Desktop/DATA202-A2/",pattern="-.*")
fullpath=file.path("/Users/Patrick/Desktop/DATA202-A2",filenames)
wcricket <- do.call("rbind",lapply(filenames,FUN=function(files){ read.csv(files)}))

nrow(wcricket)
```

```
## [1] 32097
```

Q3. (8 Marks) The dates in the innings.csv file are stored as characters, but are inconsistent. The ODI entries have a different format to the others.

a.Create a new column innings.date which has a single consistent format.

ANS:

```
fmts<-c("%Y-%m-%d","%d/%m/%Y")
# 1970-01-01 is standard of Unix/POSIX time.
innings$innings.date <- as.Date(as.numeric(apply(outer(innings$date, fmts, as.Date), 1, na.omit)), "197
innings$innings.date <- as.character(innings$innings.date)
head(innings)
```

```
##   innings.id team.id team.id.opp ground.id type       date innings.date
## 1      79650     654         231      9264  ODI 24/02/2019   2019-02-24
## 2      37696     230         231      9264  ODI 28/02/2006   2006-02-28
## 3      21153     230         231      9264 Test 2006-02-18   2006-02-18
## 4      87746     231         654      9264  ODI 03/02/1996   1996-02-03
## 5      44726     175         231      9264 Test 1984-12-21   1984-12-21
## 6      13304     231         654      9264  ODI 10/02/2010   2010-02-10
```

b.Create another column innings.Rdate which has these dates stored in the R date type. **ANS:**

```
fmts<-c("%Y-%m-%d","%d/%m/%Y")
# 1970-01-01 is standard of Unix/POSIX time.
innings$innings.Rdate <- as.Date(as.numeric(apply(outer(innings$date, fmts, as.Date), 1, na.omit)), "19
head(innings)
```

```
##   innings.id team.id team.id.opp ground.id type       date innings.date
## 1      79650     654         231      9264  ODI 24/02/2019   2019-02-24
## 2      37696     230         231      9264  ODI 28/02/2006   2006-02-28
## 3      21153     230         231      9264 Test 2006-02-18   2006-02-18
## 4      87746     231         654      9264  ODI 03/02/1996   1996-02-03
## 5      44726     175         231      9264 Test 1984-12-21   1984-12-21
## 6      13304     231         654      9264  ODI 10/02/2010   2010-02-10
##   innings.Rdate
## 1    2019-02-24
## 2    2006-02-28
## 3    2006-02-18
## 4    1996-02-03
## 5    1984-12-21
## 6    2010-02-10
```

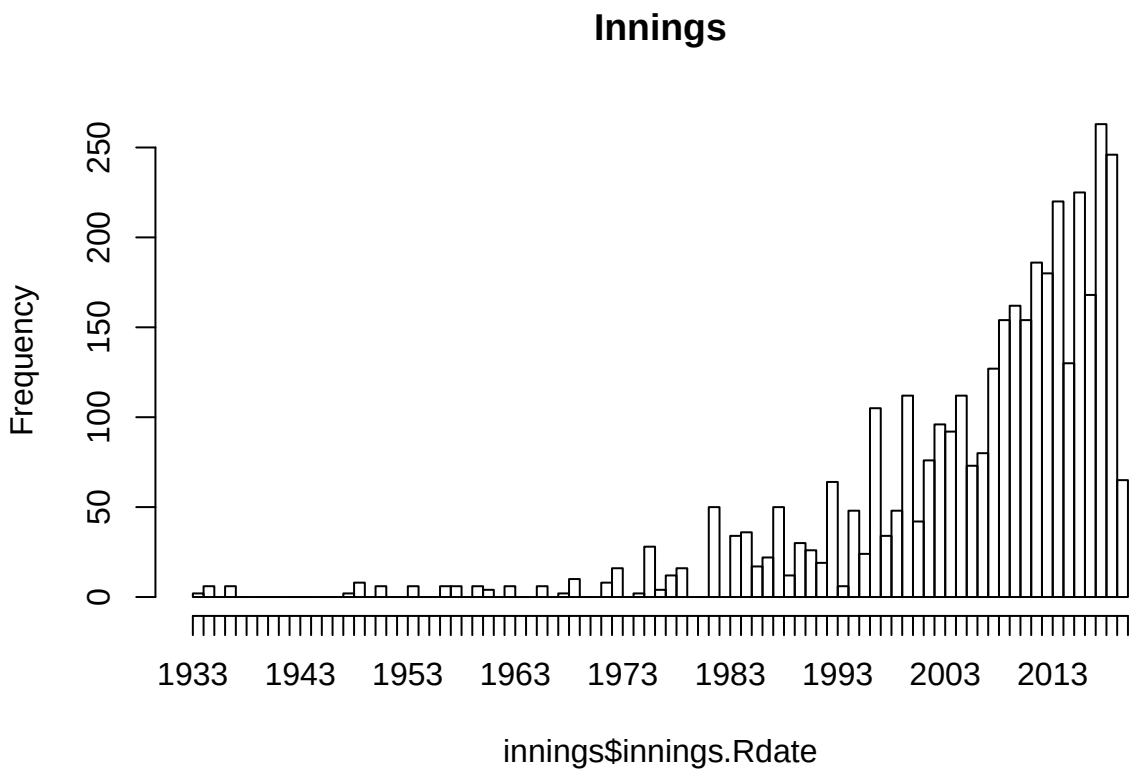
c.Find the range of dates of the innings in the database

```
range(innings$innings.Rdate, na.rm=TRUE)
```

```
## [1] "1934-12-28" "2020-03-08"
```

d.Draw a histogram of the dates of the innings recorded in the database.

```r
hist(innings$innings.Rdate, breaks="years", freq=TRUE, format="%Y", main="Innings")
```



Q4. (2 Mark) Write R code to copy the tables grounds, teams, players, innings and the combined outcomes table into your SQL database. List the tables in the database to verify that the tables have been created.

ANS:

```r
library(DBI)
library(RSQLite)
test_conn <- dbConnect(RSQLite::SQLite(), "cricket.sqlite")
dbWriteTable(test_conn, "grounds", grounds, overwrite=TRUE)
dbWriteTable(test_conn, "innings", innings, overwrite=TRUE)
dbWriteTable(test_conn, "players", players, overwrite=TRUE)
dbWriteTable(test_conn, "teams", teams, overwrite=TRUE)
dbWriteTable(test_conn, "wcricket", wcricket, overwrite=TRUE)

dbListTables(test_conn)
```

```
## [1] "grounds"  "innings"  "players"  "teams"    "types"    "wcricket"
```

Q5. (6 Marks) Using a CREATE TABLE command followed by an INSERT command, create a table called types in the database which can be used to decode the type column from its values ODI, Test, T20 into fuller descriptions One Day International, Test Match and Twenty20. Make the type column the primary key of the table. Use a SELECT command to show the contents of the table.

ANS:

```
DROP TABLE IF EXISTS types;
```

```
CREATE TABLE types (
type TEXT,
type_description TEXT,
PRIMARY KEY(type)
)
```

```
INSERT INTO types (type, type_description)
VALUES
("ODI","One Day International"),
("Test","Test Match"),
("T20","Twenty20")
```

```
SELECT * FROM types
```

Table 1: 3 records

type	type_description
ODI	One Day International
Test	Test Match
T20	Twenty20

Q6. (2 Marks)

a. What is the function of having a primary key in a table in a database?

ANS: It helps you link your table to other tables (relationships) using primary key as links.

b.What does it mean for a column to be a foreign key in a table in a database?

ANS: This referential link helps to match the foreign key column data with the data of the referenced table data. The referenced table is called the parent table and the table that involves a foreign key is called the child table.

Q7. (4 Marks) Write SQL code to count the number of innings by type of match - show your output. Write the query two ways: one way just using the innings table, and a second way using the join to types to show the full description of the type column.

ANS:

```
SELECT type,COUNT(*)
FROM Innings
GROUP BY type
```

Table 2: 3 records

type	COUNT(*)
ODI	2306
T20	1171
Test	279

```
SELECT type_description,COUNT(type_description)
FROM Innings LEFT JOIN types
ON Innings.type = types.type
GROUP BY type_description
```

Table 3: 3 records

type_description	COUNT(type_description)
One Day International	2306
Test Match	279
Twenty20	1171

Q8. (1 Mark) Write SQL code to show the maximum number of runs scored by any player in an innings.

ANS:

```
SELECT Wcricket."innings.id",Wcricket."player.id",MAX(innings_runs_scored_num)
FROM Wcricket
```

Table 4: 1 records

innings.id	player.id	MAX(innings_runs_scored_num)
27408	201658	242

Q9. (2 Marks) Write SQL code to find players and innings where 200 or more runs were scored. Show the player id, innings id, and number of runs in your output. Order by DESCENDING number of runs scored.

ANS:

```
SELECT Wcricket."innings.id",Wcricket."player.id",innings_runs_scored_num
FROM Wcricket
WHERE innings_runs_scored_num >= 200
GROUP BY innings_runs_scored_num
ORDER BY innings_runs_scored_num DESC
```

Table 5: 8 records

innings.id	player.id	innings_runs_scored_num
27408	201658	242
76485	341870	232
56378	780038	229
37634	142683	214
63131	711527	213
60793	103743	209
37532	440400	204
86235	587556	200

Q10. (3 Marks) Modify the output of the previous question to include, in addition to the number of runs scored, the name of the player, the name of the team, the date and the name of the ground where the match took place.

ANS:

```
SELECT "ground.name","player.name","team.name",innings_runs_scored_num
FROM Wcricket
LEFT JOIN Players
ON Wcricket."player.id" = Players."player.id"
LEFT JOIN Teams
ON Wcricket."team.id" = Teams."team.id"
LEFT JOIN Innings
ON Wcricket."innings.id" = Innings."innings.id"
LEFT JOIN Grounds
ON Innings."ground.id" = Grounds."ground.id"
WHERE innings_runs_scored_num >= 200
ORDER BY innings_runs_scored_num DESC
```

Table 6: 9 records

ground.name	player.name	team.name	innings_runs_scored_num
Karachi	Kiran Baluch	Pakistan	242
Dublin	AC Kerr	New Zealand	232
Mumbai	BJ Clark	Australia	229
Taunton	M Raj	India	214
Sydney	EA Perry	Australia	213
Leeds	KL Rolton	Australia	209
Shenley	MAJ Goszko	Australia	204
Scarborough	KE Flavell	New Zealand	204
Guildford	J Broadbent	Australia	200

Q11. (2 Marks) Give the names of the players who have scored more than 6000 runs in total.

ANS:

```
SELECT "player.name" ,innings_runs_scored_num
FROM (SELECT "player.name",SUM(innings_runs_scored_num) AS innings_runs_scored_num
FROM Wcricket
LEFT JOIN Players
ON Wcricket."player.id" = Players."player.id"
GROUP BY "player.name"
ORDER BY innings_runs_scored_num DESC
)
WHERE innings_runs_scored_num > 6000
```

Table 7: 7 records

player.name	innings_runs_scored_num
CM Edwards	9888
M Raj	9849
SW Bates	7745
SR Taylor	7738
MM Lanning	6540
SJ Taylor	6429
KL Rolton	6031

Q12. (2 Marks) Calculate the average number of runs per player per innings for the three types of game. Include the type_description in your output as well as the short type code. (Hint: the function AVG() computes the mean of a set of numeric values in a column.)

ANS:

```
SELECT "player.name",AVG(wcricket.innings_runs_scored_num) AS AVG,innings.type,types.type_description

FROM wcricket
LEFT JOIN players
ON players."player.id" = wcricket."player.id"

LEFT JOIN innings
ON wcricket."innings.id" = innings."innings.id"

LEFT JOIN types
ON types.type = innings.type

GROUP BY players."player.name"
```

Table 8: Displaying records 1 - 10

player.name	AVG	type	type_description
A Aitken-Drummond	3.500000	T20	Twenty20
A Battenberg-Venturini	2.789474	ODI	One Day International
A Beggs	0.000000	ODI	One Day International
A Bosch	9.333333	ODI	One Day International
A Brindle	22.032787	ODI	One Day International
A Chopra	25.614286	ODI	One Day International
A Das	2.687500	ODI	One Day International
A Elder	21.500000	ODI	One Day International
A Fellows	16.500000	ODI	One Day International
A Fujishiro	1.500000	ODI	One Day International

Q13. (2 Marks) Write R code that computes the number of seconds that New Zealand was in Level 4 lockdown earlier this year. Show the R code that creates the date-time objects needed, and the calculation of the time difference.

COVID-19 Alert Level 3 came into force at 1:30pm Monday 23 March 2020. COVID-19 Alert Level 4 came into force at 11:59pm Wednesday 25 March 2020.

```
s <- c("23 4 2020 13:30:00", "25 4 2020 23:59:00")
st <- as.POSIXct(s, format="%d %m %Y %H:%M:%S")
```

```r
difftime(st[2],st[1],units = "secs")
```

```
## Time difference of 210540 secs
```