Information Systems, Fall 2025

Homework 3.

Due November 2, 11:59 pm.

This homework is done using R/R-Studio**.**

Open the script containing the examples from the class of October 20.

Run the script (no need to install/update the library if you did it before).

Questions 1 and 2 use REGULAR periods, as in the class example!

Question 1 (5 points): Using existing variables from the class example, add the following:

a. Find shots against/shots for ratio **per team (different from class example! There we calculated it by team in each period played, now we want just totals).** Only include the code (no table) in the answer. (4 points)

```
53
54  # Part a: Find shots against/shots for ratio per team (totals, not by period)
55
56  # total shots FOR each team (across all games and periods)
57  Total_Shots_For <- sqldf("SELECT team_id_for, SUM(Shots) AS Total_Shots_For
58                            FROM ShotsTable
59                            GROUP BY team_id_for")
60
61  # total shots AGAINST each team (across all games and periods)
62  Total_Shots_Against <- sqldf("SELECT team_id_against, SUM(Shots) AS Total_Shots_Against
63                                FROM ShotsTableAgainst
64                                GROUP BY team_id_against")
65
66  # calculate ratio per team (shots against / shots for)
67  Shots_Ratio_Per_Team <- sqldf("SELECT
68                                 a.team_id_against AS team_id,
69                                 a.Total_Shots_Against,
70                                 f.Total_Shots_For,
71                                 CAST(a.Total_Shots_Against AS FLOAT)/(f.Total_Shots_For AS Ratio)
72                                 FROM Total_Shots_Against AS a
73                                 JOIN Total_Shots_For AS f
74                                 ON a.team_id_against = f.team_id_for")
75
```

b. Report the average ratio for all teams (don't round) (1 point)

```
76
77  # Part b: Report the average ratio for all teams
78  Avg_Ratio_All_Teams <- mean(Shots_Ratio_Per_Team$Ratio)
79  print(Avg_Ratio_All_Teams)
80
```

Avg_Ratio_All_Teams	1.00401942471399

Question 2 (10 points):

a. Similarly to the class example, construct tables containing average **missed shots** (the event name is 'Missed Shot') per team per period, both for and against. Only include your code in the answer. (7 points).

```
# Part a: Construct tables with average missed shots per team per period

# missed shots FOR each team per period
MissedShotsTableFor <- sqldf("SELECT game_id, period, team_id_for, COUNT(event) AS MissedShots
                              FROM NHL_Data
                              WHERE event = 'Missed Shot' AND periodType = 'REGULAR'
                              GROUP BY game_id, period, team_id_for")

# average missed shots FOR each team per period
MissedShots_Avg_by_Team_For <- sqldf("SELECT team_id_for, AVG(MissedShots) AS MissedShotAvg
                                      FROM MissedShotsTable
                                      GROUP BY team_id_for")

# missed shots AGAINST each team per period
MissedShotsTableAgainst <- sqldf("SELECT game_id, period, team_id_against, COUNT(event) AS MissedShots
                                  FROM NHL_Data
                                  WHERE event = 'Missed Shot' AND periodType = 'REGULAR'
                                  GROUP BY game_id, period, team_id_against")

# average missed shots AGAINST each team per period
MissedShots_Avg_by_Team_Against <- sqldf("SELECT team_id_against, AVG(MissedShots) AS MissedShotAvg
                                          FROM MissedShotsTableAgainst
                                          GROUP BY team_id_against")
```

b. (3 points): Using both average missed and on target shots tables, construct the table with average ratio of missed/on target for each team per period. Report the average ratio (the average of the resulting table). Answers must include both the code and the number. Don't round. (Hint: you don't need both 'for' and 'against', only 'for')

```
# Part b: Construct table with average ratio of missed/on target for each team per period

# using the existing ShotsTable (on target shots) and new MissedShotsTable
Missed_OnTarget_Ratio <- sqldf("SELECT
                                m.team_id_for AS team_id,
                                AVG(CAST(m.MissedShots AS FLOAT) / s.Shots) AS Avg_Ratio
                                FROM MissedShotsTable AS m
                                JOIN ShotsTable AS s
                                ON m.game_id = s.game_id
                                AND m.period = s.period
                                AND m.team_id_for = s.team_id_for
                                GROUP BY m.team_id_for")

# report the average ratio (average of the resulting table)
Overall_Avg_Ratio <- mean(Missed_OnTarget_Ratio$Avg_Ratio)
print(Overall_Avg_Ratio)
```

	team_id	Avg_Ratio
1	1	0.5202265
2	2	0.4858234
3	3	0.5069381
4	4	0.5114625
5	5	0.4610003
6	6	0.4968153
7	7	0.4694258
8	8	0.5011819
9	9	0.4943740
10	10	0.5464612
11	12	0.5311794
12	13	0.4725858
13	14	0.5418471
14	15	0.5606436
15	16	0.4484470
16	17	0.5107693
17	18	0.5086924
18	19	0.5332622
19	20	0.5362893
20	21	0.4863753
21	22	0.5072260
22	23	0.5136149
23	24	0.5627286
24	25	0.5706031
25	26	0.5856334
26	27	0.5221498
27	28	0.5372536
28	29	0.4942859
29	30	0.4969430
30	52	0.5175179
31	53	0.5352946
32	54	0.5297234

For Question 3 use ALL Periods, not just regular!

Question 3 (10 points + 1 extra):

a. Construct tables containing total goals for and against each team (the event name is 'Goal') with team id (you need two tables, or may join them if you know how). Only show the code. (7 points)

```
132
133 # Part a: Construct tables containing total goals for and against each team
134
135 # total goals FOR each team (across all periods)
136 Goals_For_Table <- sqldf("SELECT team_id_for, COUNT(event) AS Total_Goals_For
137                           FROM NHL_Data
138                           WHERE event = 'Goal'
139                           GROUP BY team_id_for")
140
141 # total goals AGAINST each team (across all periods)
142 Goals_Against_Table <- sqldf("SELECT team_id_against, COUNT(event) AS Total_Goals_Against
143                               FROM NHL_Data
144                               WHERE event = 'Goal'
145                               GROUP BY team_id_against")
146
147
```

b. Find goal differences for each team (for – against), total number of goals *(R function is sum(vector))*, and the average goal difference. Report the code, total goals, and the average (not the table with differences). (3 points)

```
147
148 # Part b: Find goal differences for each team (for - against)
149
150 # join the two tables and calculate goal difference
151 Goal_Differences <- sqldf("SELECT
152                            f.team_id_for AS team_id,
153                            f.Total_Goals_For,
154                            a.Total_Goals_Against,
155                            (f.Total_Goals_For - a.Total_Goals_Against) AS Goal_Difference
156                            FROM Goals_For_Table AS f
157                            JOIN Goals_Against_Table AS a
158                            ON f.team_id_for = a.team_id_against")
159
160 # total number of goals for each team using R function sum(vector)
161 Total_Goals <- Goals_For_Table$Total_Goals_For
162 print(Total_Goals)
163
164 # calculate average goal difference
165 Avg_Goal_Difference <- mean(Goal_Differences$Goal_Difference)
166 print(Avg_Goal_Difference)
167
```

```
> print(Total_Goals)
 [1] 1120 1406 1512 1312 1715 1504 1050 1336 1353 1360 1160 1235 1569 1598 1589 1284 1449 1450 1273 1269 1246 1172 1537 1391 1370  355 1502
[28] 1341 1404 1394  773  327
```

Avg_Goal_Difference	0

c. Extra up to 2 points. Does your observed average goal difference make sense? Explain why (think of a 'league' with two teams that play each other once).

Yes, the observed average goal difference makes sense. This is because all of the goals scored BY one team (positive number) is the same as all of the goals scored AGAINST one team (negative number). Since every team in the dataset plays against one another, all of these numbers cancel out to zero eventually.