HW: Week 15

36-350 – Statistical Computing

Week 15 – Fall 2020

Name: Kimberly Zhang

Andrew ID: kyz

You must submit **your own** lab as a knitted PDF file on Gradescope.

It will help you to have "correct" solutions for Q6-Q8 from HW 14 in order to do HW 15. (Your solutions may differ but if they get the same result, they are of course fine.) I show them below:

Q6:

```
create table new_table
  (player integer references more_player_stats, prl numeric, position text);
insert into new_table (player, prl)
  (select id, round(per - 67*va/(gp * minutes),1) from more_player_stats);
update new_table set position = 'PF' where prl >= 11.3;
update new_table set position = 'PG' where prl >= 10.8 and prl < 11.3;
update new_table set position = 'C'  where prl >= 10.6 and prl < 10.8;
update new_table set position = 'SG/SF' where prl < 10.6;
```

Q7:

```
alter table player_bios add column position text;
update player_bios set position = new_table.position from new_table where id = new_table.player;
```

Q8:

```
alter table player_bios add column height_inches numeric;
update player_bios
   set height_inches = 12*split_part(height,'-',1)::integer + split_part(height,'-',2)::integer;
alter table player_bios drop column height;
alter table player_bios rename column height_inches to height;
```

Run the above chunks of `postgres` code prior to starting Q1 below.

Question 1

(15 points)

Create a table called `players` with the following columns:

- `id`: a serial primary key
- `firstname` and `lastname`: both text
- `position`: a string as defined in HW 14

- `age`, `height` (in inches, as determined in HW 14), and `weight`: all as integers with checks that the values are greater than zero
- `college`, `country`, `draft_year`, `draft_round`, and `draft_number`: all as text

These data are available in `player_bios`. Use `create` and `insert` with `select` to copy these data to `players`. Show the first five first and last names.

```
postgres=# create table players (
id serial primary key, firstname text, lastname text, position text,
age integer check(age > 0), height integer check(height > 0),
weight integer check(weight > 0), college text, country text, draft_year text, draft_round text, draft_n
CREATE TABLE
postgres=# insert into players (firstname, lastname, position, age, height,
weight, college, country, draft_year, draft_round, draft_number)
(select firstname, lastname, position, age, height, weight, college, country,
draft_year, draft_round, draft_number from player_bios);
INSERT 0 476
postgres=# select * from players limit 5;
 id | firstname | lastname  | position | age | height | weight |        college         | country | dra
----+-----------+-----------+----------+-----+--------+--------+------------------------+---------+----
  1 | Andre     | Miller    | PG       |  40 |     75 |    200 | Utah                   | USA     | 199
  2 | Andrew    | Goudelock | PG       |  27 |     75 |    200 | College of Charleston  | USA     | 201
  3 | Austin    | Rivers    | PG       |  23 |     76 |    200 | Duke                   | USA     | 201
  4 | Avery     | Bradley   | PG       |  25 |     74 |    180 | Texas                  | USA     | 201
  5 | Brandon   | Jennings  | PG       |  26 |     73 |    169 | None                   | USA     | 200
(5 rows)
```

Question 2

(15 points)

Now you will construct a table of player statistics for the 2015-16 NBA season, called `stats`. The columns:

- `id`: a serial primary key
- `player`: a foreign integer key that references the `players` table (when inserting, you would select players.id as player)
- `team`: a `char(3)`
- `gp`: an integer
- `minutes`, `fga`, `fgpct`, `tppct`, `fta`, `ftpct`, `ast`, `tov`, `pts`, `tovr`, `ewa`: all numeric

Some of these quantities are in `player_stats`, some are in `more_player_stats`, and some are in `players`. To build the table, you will chain two `join`s together: you will join `player_stats` and `players` using first *and* last names, and further join `more_player_states` using id's.

When you are all done, display the first three rows of your new table.

```
postgres=# create table stats (id serial primary key,
player integer references players (id), team char(3), gp integer,
minutes numeric, fga numeric, fgpct numeric, tppct numeric, fta numeric,
ftpct numeric, ast numeric, tov numeric, pts numeric, tovr numeric,
ewa numeric);
CREATE TABLE
postgres=# insert into stats (player, team, gp, minutes, fga, fgpct, tppct,
fta, ftpct, ast, tov, pts, tovr, ewa)
(select p.id, ps.team, ps.gp, ps.minutes, ps.fga, ps.fgpct, ps.tppct, ps.fta,
ps.ftpct, ps.ast, ps.tov, ps.pts, mps.tovr, mps.ewa
from player_stats ps join players p on ps.firstname = p.firstname
and ps.lastname = p.lastname join more_player_stats mps on ps.id = mps.id);
```

```
INSERT 0 476
postgres=# select * from stats limit 3;
 id  | player | team | gp | minutes | fga | fgpct | tppct | fta | ftpct | ast | tov | pts | tovr | ewa
-----+--------+------+----+---------+-----+-------+-------+-----+-------+-----+-----+-----+------+-----
 477 |    412 | CHI  | 69 |    16.1 | 6.8 |  40.1 |  35.7 | 0.9 |  76.6 | 2.6 | 1.2 | 7.1 | 10.8 |  0.5
 478 |    413 | ORL  | 78 |    23.9 | 7.4 |  47.3 |  29.6 | 2.5 |  66.8 | 1.6 | 0.8 | 9.2 |  7.7 |  5.2
 479 |     66 | CHA  | 21 |     4.4 | 0.9 |  26.3 |  30.0 | 0.6 |  41.7 | 0.1 | 0.2 | 0.9 | 13.2 | -0.3
(3 rows)
```

Question 3

(10 points)

List the names and average points per game of the top 25 scorers, in descending order of average points per game, restricting your attention to those players who averaged at least eight minutes per game and who played at least 10 games.

```
postgres=# select firstname, lastname, pts
from players p join stats s on p.id = s.player
where minutes>=8 and gp >=10
order by pts desc
limit 25;
 firstname | lastname  | pts
-----------+-----------+------
 Stephen   | Curry     | 30.1
 James     | Harden    | 29.0
 Kevin     | Durant    | 28.2
 DeMarcus  | Cousins   | 26.9
 LeBron    | James     | 25.3
 Damian    | Lillard   | 25.1
 Anthony   | Davis     | 24.3
 DeMar     | DeRozan   | 23.5
 Russell   | Westbrook | 23.5
 Paul      | George    | 23.1
 Isaiah    | Thomas    | 22.2
 Klay      | Thompson  | 22.1
 Carmelo   | Anthony   | 21.8
 Blake     | Griffin   | 21.4
 Kawhi     | Leonard   | 21.2
 Kyle      | Lowry     | 21.2
 Kemba     | Walker    | 20.9
 Jimmy     | Butler    | 20.9
 C.J.      | McCollum  | 20.8
 Andrew    | Wiggins   | 20.7
 Brook     | Lopez     | 20.6
 Eric      | Bledsoe   | 20.4
 John      | Wall      | 19.9
 Gordon    | Hayward   | 19.7
 Kyrie     | Irving    | 19.6
(25 rows)
```

Question 4

(10 points)

Rank teams according to average points per game over all players on the team who played at least 24 minutes per game on average. Round the output average to two decimal places.

```
postgres=# select team, round(avg(pts),2) as "avg pts" from stats
where minutes>=24 group by team order by avg(pts) desc;
 team | avg pts
------+---------
 GSW  |   16.98
 OKC  |   16.42
 HOU  |   16.33
 LAC  |   15.97
 DET  |   15.66
 POR  |   15.10
 BKN  |   14.93
 CHI  |   14.84
 MIN  |   14.64
 UTA  |   14.60
 LAL  |   14.58
 NOP  |   14.58
 IND  |   14.58
 BOS  |   14.24
 ORL  |   14.14
 SAC  |   14.00
 TOR  |   13.98
 PHX  |   13.80
 ATL  |   13.76
 MIL  |   13.46
 CLE  |   13.43
 SAS  |   13.38
 DEN  |   13.20
 CHA  |   12.89
 WAS  |   12.80
 DAL  |   12.78
 MEM  |   12.47
 NYK  |   12.40
 MIA  |   12.27
 PHI  |   11.63
(30 rows)
```

Question 5

(10 points)

List the top five players (in order last name, then first name) with the highest minutes per game but a negative "estimated wins added" to the team (`ewa`).

```
postgres=# select lastname, firstname
from players p join stats s on p.id = s.player
where ewa < 0
order by minutes desc
limit 5;
 lastname | firstname
----------+-----------
 Mudiay   | Emmanuel
 Korver   | Kyle
 Winslow  | Justise
```

```
 Thompson | Hollis
 Waiters  | Dion
(5 rows)
```

Question 6

(10 points)

List the last name, first name, position, points per game, and the average points per game of players with the same position in each row. Show the results in order of position first, and then by descending points per game. Computing the average described above requires the use of an `over` clause and a `partition by` window function: put `over (partition by players.position)` before the `from` clause. Round the average points to two decimal places. Limit your query to players who play more than 24 minutes per game. Show the first 10 rows of your resulting table. The fifth column will all have the same numbers: this is, e.g., the average points per game for all players listed as centers. (That's what it is in my output: I get an average of 13.27 points per game for centers.)

```
postgres=# select lastname, firstname, position, pts,
round(avg(pts) over (partition by p.position),2) as "avg_pts"
from players p join stats s on p.id = s.player
where minutes > 24
order by position, pts desc
limit 10;
 lastname |  firstname   | position | pts  | avg_pts
----------+--------------+----------+------+---------
 Cousins  | DeMarcus     | C        | 26.9 |   13.27
 Lopez    | Brook        | C        | 20.6 |   13.27
 Towns    | Karl-Anthony | C        | 18.3 |   13.27
 Vucevic  | Nikola       | C        | 18.2 |   13.27
 Okafor   | Jahlil       | C        | 17.5 |   13.27
 Gasol    | Marc         | C        | 16.6 |   13.27
 Gasol    | Pau          | C        | 16.5 |   13.27
 Drummond | Andre        | C        | 16.2 |   13.27
 Monroe   | Greg         | C        | 15.3 |   13.27
 Horford  | Al           | C        | 15.2 |   13.27
(10 rows)
```

Question 7

(10 points)

List the last name, first name, field-goal percentage (`fgpct`), free-throw percentage (`ftpct`), and three-point-field-goal percentage (`tppct`) for all players who, over the course of the season, had a field-goal percentage of at least 50, a free-throw percentage of at least 80, and a three-point-field-goal percentage of at least 35. Order your output by field-goal percentage, decreasing.

```
postgres=# select lastname, firstname, fgpct, ftpct, tppct
from players p join stats s on p.id = s.player
where fgpct >= 50 and ftpct >= 80 and tppct >= 35
order by fgpct;
 lastname | firstname | fgpct | ftpct | tppct
----------+-----------+-------+-------+-------
 Curry    | Stephen   |  50.4 |  90.8 |  45.4
 Durant   | Kevin     |  50.5 |  89.8 |  38.7
 Leonard  | Kawhi     |  50.6 |  87.4 |  44.3
(3 rows)
```

Question 8

(10 points) The turnover ratio column is defined as (`tov` x 100) divided by [`fga`+(`fta` x 0.44)+`ast`+`tov`]. Compute this quantity directly given your `stats` table and show that the computed values are essentially the same as those stored in the column `tovr`. (To be clear: this means to display player id, the computed column (rounded to two decimal places), and the value of `tovr`. Show your first 10 results.)

The tricky part here is avoiding dividing by zero. You want to perform the calculation for those rows where, e.g., the player is "not in (...)", where the part inside the parentheses is a selection of players where the statistics in the denominator are each equal to zero.

```
postgres=# select player, round((tov*100)/(fga + (fta * 0.44) + ast + tov), 2)
as "computed turnover", tovr
from stats where player not in
(select player from stats where (fga + (fta*0.44)+ast+tov) = 0)
limit 10;
 player | computed turnover | tovr
--------+-------------------+------
    412 |             10.91 | 10.8
    413 |              7.34 |  7.7
     66 |             13.66 | 13.2
    185 |             16.20 | 16.3
     34 |              7.22 |  7.3
     35 |              5.02 |  5.2
     67 |             11.69 | 11.5
     68 |              3.16 |  2.5
     36 |             14.58 | 14.6
     69 |              9.77 | 10.0
(10 rows)
```

Question 9

(10 points)

One thing we did not cover in the notes is the concept of "Common Table Expressions," known informally as `with` clauses. Here's a simple example:

```
with discrete_temp (max_temp,temp_group) as
  (select max_temp,
    case when max_temp >= 90 then 'hot'
         when max_temp <= 50 then 'cold'
         else 'alright'
    end
  from temperature_readings)
  select
    temp_group,count(*)
    from discrete_temp
    group by temp_group
    order by count(*) desc;
```

The `with` clause creates a temporary table (here, `discrete_temp`) that is then used in the subsequent selection. The output is a table showing the number of hot, cold, and alright days. Using a `with` clause is simply a variation on using sub-queries (i.e., `select`s within `select`s).

Use a `with` clause below to create three assists groups (`high` for a player with more than 2 assists per game, `low` for a player with less than 1 assist per game, and `medium` otherwise), then output the number of players in each group (i.e., output the group name and the count as two columns).

```
postgres=# with discrete_ast (ast, ast_group) as
(select ast,
case when ast > 2 then 'high'
when ast < 1 then 'low'
else 'medium' end from stats)
select ast_group, count(*)
from discrete_ast
group by ast_group
order by count(*) desc;
 ast_group | count
-----------+-------
 low       |   175
 medium    |   158
 high      |   143
(3 rows)
```

And with that, you are done.