SQL HW 2

Prabha Nayak

April 23, 2018

```
## Loading required package: RPostgreSQL
```

```
## Warning: package 'RPostgreSQL' was built under R version 3.4.4
```

```
## Loading required package: DBI
```

```
## Warning: package 'DBI' was built under R version 3.4.4
```

1. CMSCLAIMS

a. For every possible value in hmo_mo, what is the average carrier reimbursement? Order by number of HMO months in ascending order.

```
SELECT hmo_mo, AVG (carrier_reimb) FROM cmsclaims WHERE hmo_mo IS NOT NULL GROUP BY (c
msclaims.hmo_mo) ORDER BY hmo_mo;
```

Displaying records 1 - 10

hmo_mo	avg
0	815.4228
1	969.5914
2	1007.2355
3	999.3674
4	1013.4261
5	993.1586
6	988.7066
7	978.4729
8	915.1374
9	920.1892

b. For every value of carrier reimbursement, how many total months of HMO coverage were provided? Order by carrier reimbursements in descending order.

```
SELECT carrier_reimb, SUM (hmo_mo) AS hmo_mos FROM cmsclaims WHERE carrier_reimb IS NO
T NULL GROUP BY (cmsclaims.carrier_reimb) ORDER BY carrier_reimb DESC LIMIT 5;
```

5 records

carrier_reimb	hmo_mos
16530	12
13040	0
12500	12
11730	0
11630	0

c. Using your query from part b, create a query that finds the largest carrier reimbursement when the total months of HMO coverage is zero.

```
SELECT MAX(carrier_reimb) AS max_reimb FROM
(SELECT carrier_reimb, SUM(hmo_mo) AS hmo_mos FROM cmsclaims GROUP BY carrier_reimb) A
S sub_q WHERE hmo_mos = 0;
```

1 records

max_reimb
13040

2 CMSPOP a. Return the number of instances of depression by sex, ordered by sex in ascending order.

```
SELECT sex, COUNT(*) FROM cmspop
WHERE depression= True
GROUP BY sex
ORDER BY sex ASC;
```

2 records

sex	count
female	235200
male	166927

b. Return the most common month to be born in.

```
SELECT mo
FROM
(SELECT mo, COUNT(*) AS month_born FROM
(SELECT *, date_part('month', dob) AS mo FROM cmspop) AS sub_sub_q
GROUP BY mo) AS sub_q
ORDER BY month_born DESC LIMIT 1;
```

1 records

mo
6

c. Return the average birth year for all records.

```
SELECT ROUND( AVG( date_part('year', dob))) FROM cmspop;
```

1 records

round
1936

d. Return the average age (should contain a decimal) of those who died under the age of 50.

```
SELECT AVG(age) FROM (SELECT *, ((dod - dob)/ 365.0) AS age FROM cmspop WHERE dod IS N
OT NULL) AS sub_q
WHERE age < 50.0;
```

1 records

avg
44.7581253615112

e. Now return the average age of those who died under the age of 50 to just two decimal places. Hint: if you get stuck, Google the error message.

```
SELECT ROUND( AVG( age)) FROM (SELECT *, ((dod - dob)/ 365.0) AS age FROM cmspop WHER
E dod IS NOT NULL) AS sub_q
WHERE age < 50.0;
```

1 records

round
45

3.

a. Return the state and number of heart failures for the state with the greates number of heart failures.

```
SELECT state, count(heart_fail) as heart_fail FROM cmspop WHERE heart_fail= True GROU
P BY state ORDER BY heart_fail DESC LIMIT 1;
```

1 records

state	heart_fail
CA	47281

b. Find which county in California has the most hispanics. Return columns for county, race, and the number of hispanics in that county. Hint: you can group by more than one column at a time.

```
SELECT * FROM (SELECT county, race, count(*) AS num_hispanic FROM cmspop WHERE state
= 'CA' GROUP BY county, race) AS sub_q
WHERE race= 'hispanic' ORDER BY num_hispanic DESC LIMIT 1;
```

1 records

county	race	num_hispanic
200	hispanic	2817

4. The data in cmspop was collected on Jan 1st, 2010. With that in mind:

a. Return the youngest age in integer years at the time of data collection by sex for those who had cancer and were alive, ordered by sex in in ascending order. To be clear, age in integer years means that age should be calculated like when you tell your age to someone: for example, “I’m 62 years old.” You don’t say, “I’m 62.7456 years old”. Hint: use the floor() function.

```
SELECT sex, floor(MIN(age)) FROM (SELECT sex, (to_date('2010-01-01', 'YYYY_MM_DD') - d
ob) :: float/365 AS age FROM cmspop WHERE dod IS NULL AND cancer = TRUE) AS sub_q GROU
P BY sex ORDER BY sex ASC;
```

2 records

sex	floor
female	26

sex	floor
male	26

b. Return the id, sex, race, and age in integer years at the time of data collection for the oldest living person in Massachusetts that has all complications (Alzheimers, depression, cancer, and heart failure).

```
SELECT id, sex, race, floor (age) AS age FROM (SELECT id, sex, race, (to_date ('2010-0
1-01', 'YYYY-MM-DD') - dob):: float/365 AS age FROM cmspop WHERE dod IS NULL AND stat
e ='MA' AND alz_rel_sen = TRUE AND depression = TRUE AND cancer = TRUE AND heart_fail
= TRUE ORDER BY age DESC LIMIT 1) AS sub_q;
```

1 records

id	sex	race	age
AB33557A62C914C7	female	white	100

5. Assume that, for the deceased, the flags for Alzheimers, depression, cancer, and heart failure indicate cause of death. Using the table cmspop, consider the following scenario:

a. In the 0th, 80th, 150th, 160th, 180th, 190th, and 220th counties within Wyoming, each of the deceased had a single cause of death. Count the number of deaths caused by each complication for those individuals. Hints: you can use parenthesis in WHERE statements to group conditions and, also in WHERE statements, you can pass a list of values to IN.

```
SELECT cause_of_death, COUNT (*) FROM (SELECT *, CASE
WHEN alz_rel_sen = TRUE THEN 'alz_rel_sen'
WHEN heart_fail = TRUE THEN 'heart_fail'
WHEN cancer = TRUE THEN 'cancer'
WHEN depression = TRUE THEN 'depression'
END AS cause_of_death FROM cmspop WHERE dod IS NOT NULL
AND (alz_rel_sen = TRUE
OR heart_fail = TRUE
OR cancer = TRUE
OR depression = TRUE)
AND state= 'WY'
AND county IN (0, 80, 150, 160, 180, 190, 220)
) AS sub_q GROUP BY cause_of_death;
```

4 records

cause_of_death	count
alz_rel_sen	2

cause_of_death	count
cancer	1
depression	1
heart_fail	4