HW4

Kailin

2024-10-27

```
library(tidyverse)
library(DBI)
library(dbplyr)
library(bigrquery)
```

```
project <- "hw-439518"
```

```
con <- dbConnect(
  bigrquery::bigquery(),
  project = "bigquery-public-data",
  dataset = "chicago_crime",
  billing = project
)
con
```

```
## <BigQueryConnection>
##   Dataset: bigquery-public-data.chicago_crime
##   Billing: hw-439518
```

```
dbListTables(con)
```

```
## ! Using an auto-discovered, cached token.

##   To suppress this message, modify your code or options to clearly consent to
##   the use of a cached token.

##   See gargle's "Non-interactive auth" vignette for more details:

##   <https://gargle.r-lib.org/articles/non-interactive-auth.html>

## i The bigrquery package is using a cached token for 'kklynxxu@gmail.com'.

## Auto-refreshing stale OAuth token.

## [1] "crime"
```

Write a first query that counts the number of rows of the `crime` table in the year 2016. Use code chunks with {sql connection = con} in order to write SQL code within the document.

```sql
SELECT count(primary_type), count(*)
FROM crime
WHERE year = 2016
LIMIT 10;
```

Table 1: 1 records

f0_	f1_
269922	269922

Next, count the number of arrests grouped by primary_type in 2016. Note that is a somewhat similar task as above, with some adjustments on which rows should be considered. Sort the results, i.e. list the number of arrests in a descending order.

```sql
SELECT primary_type, COUNT(*) AS arrest_count
FROM crime
WHERE year = 2016 AND arrest = TRUE
GROUP BY primary_type
ORDER BY arrest_count DESC
LIMIT 10;
```

Table 2: Displaying records 1 - 10

primary_type	arrest_count
NARCOTICS	13327
BATTERY	10333
THEFT	6522
CRIMINAL TRESPASS	3724
ASSAULT	3492
OTHER OFFENSE	3415
WEAPONS VIOLATION	2511
CRIMINAL DAMAGE	1669
PUBLIC PEACE VIOLATION	1116
MOTOR VEHICLE THEFT	1098

We can also use the date for grouping. Count the number of arrests grouped by hour of the day in 2016. You can extract the latter information from date via EXTRACT(HOUR FROM date). Which time of the day is associated with the most arrests?
19:00 is associated with the most arrests.

```sql
SELECT EXTRACT(HOUR FROM date) AS hour_of_day, COUNT(*) AS arrest_count
FROM crime
WHERE year = 2016 AND arrest = TRUE
GROUP BY hour_of_day
ORDER BY arrest_count DESC
LIMIT 10;
```

Table 3: Displaying records 1 - 10

hour_of_day	arrest_count
19	3843
18	3481
20	3302
21	2961
16	2933
22	2896
11	2895
17	2820
12	2787
14	2774

Focus only on HOMICIDE and count the number of arrests for this incident type, grouped by year. List the results in descending order.

```
SELECT year, COUNT(*) AS arrest_count
FROM crime
WHERE primary_type = 'HOMICIDE' AND arrest = TRUE
GROUP BY year
ORDER BY arrest_count DESC
LIMIT 10;
```

Table 4: Displaying records 1 - 10

year	arrest_count
2001	430
2002	427
2003	382
2020	349
2022	306
2004	294
2021	292
2016	289
2008	287
2006	284

Find out which districts have the highest numbers of arrests in 2015 and 2016. That is, count the number of arrests in 2015 and 2016, grouped by year and district. List the results in descending order. District 11 has the highest numbers of arrests in 2015 and 2016.

```
SELECT year, district, COUNT(*) AS arrest_count
FROM crime
WHERE year IN (2015, 2016) AND arrest = TRUE
GROUP BY year, district
ORDER BY arrest_count DESC
LIMIT 10;
```

Table 5: Displaying records 1 - 10

year	district	arrest_count
2015	11	8974
2016	11	6575
2015	7	5549
2015	15	4514
2015	6	4474
2015	25	4450
2015	4	4325
2015	8	4113
2016	7	3655
2015	10	3622

Lets switch to writing queries from within R via the DBI package. Create a query object that counts the number of arrests grouped by primary_type of district 11 in year 2016. The results should be displayed in descending order.

```
query <- "SELECT primary_type, COUNT(*) AS arrest_count
          FROM crime
          WHERE year = 2016 AND district = 11 AND arrest = TRUE
          GROUP BY primary_type
          ORDER BY arrest_count DESC"
result1 <- dbGetQuery(con, query)
print(head(result1, 10))
```

```
## # A tibble: 10 x 2
##    primary_type                     arrest_count
##    <chr>                                   <int>
##  1 NARCOTICS                                3634
##  2 BATTERY                                   635
##  3 PROSTITUTION                              511
##  4 WEAPONS VIOLATION                         303
##  5 OTHER OFFENSE                             255
##  6 ASSAULT                                   206
##  7 CRIMINAL TRESPASS                         205
##  8 PUBLIC PEACE VIOLATION                    135
##  9 INTERFERENCE WITH PUBLIC OFFICER          119
## 10 CRIMINAL DAMAGE                           106
```

Try to write the very same query, now using the dbplyr package. For this, you need to first map the crime table to a tibble object in R.

```
cri <- tbl(con, "crime")
```

Again, count the number of arrests grouped by primary_type of district 11 in year 2016, now using dplyr syntax.

```
query2 <- cri %>% select(primary_type, year, district, arrest) %>%
  filter(year == 2016 & district == 11 & arrest == TRUE)%>%
  group_by(primary_type) %>%
```

```
  summarise(total = n())%>%
  collect()
print(head(query2, 10))
```

```
## # A tibble: 10 x 2
##    primary_type           total
##    <chr>                  <int>
##  1 HOMICIDE                  28
##  2 DECEPTIVE PRACTICE        63
##  3 ASSAULT                  206
##  4 WEAPONS VIOLATION        303
##  5 BURGLARY                  22
##  6 PROSTITUTION             511
##  7 OTHER OFFENSE            255
##  8 THEFT                     98
##  9 PUBLIC PEACE VIOLATION   135
## 10 GAMBLING                  32
```

Count the number of arrests grouped by primary_type and year, still only for district 11. Arrange the result by year.

```
query3 <- cri %>% select(primary_type, year, district, arrest) %>%
  filter(district == 11 & arrest == TRUE) %>%
  group_by(primary_type, year) %>%
  summarise(total = n())%>%
  arrange(year) %>%
  collect()
```

```
## `summarise()` has grouped output by "primary_type". You can override using the
## `.groups` argument.
```

```
print(head(query3, 10))
```

```
## # A tibble: 10 x 3
## # Groups:   primary_type [10]
##    primary_type                year total
##    <chr>                      <int> <int>
##  1 CRIM SEXUAL ASSAULT         2001    17
##  2 BATTERY                     2001   962
##  3 DECEPTIVE PRACTICE          2001    84
##  4 PROSTITUTION                2001   424
##  5 GAMBLING                    2001    71
##  6 THEFT                       2001   419
##  7 INTIMIDATION                2001     3
##  8 OTHER OFFENSE               2001   266
##  9 OFFENSE INVOLVING CHILDREN  2001    44
## 10 CRIMINAL DAMAGE             2001   163
```

```
dbDisconnect(con)
```