Assignment 4

Aulia Dini

GitHub Link : https://github.com/aulia0716/Assignment_4_727.git

Install Library

After you have initialized a project, paste your project ID into the following chunk.

```r
project <- "refined-circuit-403104"
```

We will connect to a public database, the Chicago crime database, which has data on crime in Chicago.

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

```
<BigQueryConnection>
  Dataset: bigquery-public-data.chicago_crime
  Billing: refined-circuit-403104
```

We can look at the available tables in this database using `dbListTables`.

Note: When you run this code, you will be sent to a browser and have to give Google permissions to Tidyverse API Packages. **Make sure you select all to give access or else your code will not run.**

```r
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 'aulia@umich.edu'.

[1] "crime"
```

Information on the ‘crime‘ table can be found here:

https://cloud.google.com/bigquery/public-data/chicago-crime-data

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.

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

Table 1: 1 records

f0_
269840

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.

```
SELECT primary_type, count(*) AS count_crime
FROM crime
WHERE year=2016 AND arrest=TRUE
GROUP BY primary_type
ORDER BY count_crime DESC
LIMIT 100;
```

Table 2: Displaying records 1 - 10

primary_type	count_crime
NARCOTICS	13327
BATTERY	10332
THEFT	6522
CRIMINAL TRESPASS	3724
ASSAULT	3492
OTHER OFFENSE	3415
WEAPONS VIOLATION	2511
CRIMINAL DAMAGE	1669
PUBLIC PEACE VIOLATION	1116
MOTOR VEHICLE THEFT	1097

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?

```sql
SELECT EXTRACT(HOUR FROM date) AS hour, COUNT(*) AS count_crime
FROM crime
WHERE EXTRACT(YEAR FROM date) = 2016 AND arrest = TRUE
GROUP BY hour
ORDER BY count_crime DESC
LIMIT 100;
```

Table 3: Displaying records 1 - 10

hour	count_crime
10	5306
11	5200
12	4941
7	4900
8	4735
9	4675
1	4288
6	4261
2	4029
3	3750

The time of the day is associated with the most arrests is 10.

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 count_crime
FROM crime
WHERE primary_type = 'HOMICIDE' and arrest = TRUE
GROUP BY year
ORDER BY count_crime DESC
LIMIT 100;
```

Table 4: Displaying records 1 - 10

year	count_crime
2001	430
2002	423
2003	379
2020	339
2004	293
2016	286
2008	286
2006	281
2005	281
2021	275

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.

```
SELECT EXTRACT(YEAR FROM date) AS  year, district, COUNT(*) AS count_crime
FROM crime
WHERE (EXTRACT(YEAR FROM date) = 2015 OR EXTRACT(YEAR FROM date) = 2016) AND arrest = TRUE
GROUP BY year, district
ORDER BY count_crime DESC
LIMIT 100;
```

Table 5: Displaying records 1 - 10

year	district	count_crime
2015	11	8974
2016	11	6575

year	district	count_crime
2015	7	5549
2015	15	4514
2015	6	4473
2015	25	4448
2015	4	4325
2015	8	4112
2016	7	3654
2015	10	3621

```sql
SELECT year, district, COUNT(*) AS count_crime
FROM crime
WHERE (year = 2015 OR year= 2016) AND arrest = TRUE
GROUP BY year, district
ORDER BY count_crime DESC
LIMIT 100;
```

Table 6: Displaying records 1 - 10

year	district	count_crime
2015	11	8974
2016	11	6575
2015	7	5549
2015	15	4514
2015	6	4473
2015	25	4448
2015	4	4325
2015	8	4112
2016	7	3654
2015	10	3621

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.

Execute the query.

```r
#install.packages("RSQLite")
```

```r
library(DBI)

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

sql <- "SELECT primary_type, count(*) AS count_arrests
        FROM crime
        WHERE year = 2016 AND district = 11 AND arrest = TRUE
        GROUP BY primary_type
        ORDER BY count_arrests DESC
        LIMIT 100"

result <- dbGetQuery(con, sql)

first_row <- head(result, 10)
first_row
```

```
# A tibble: 10 x 2
   primary_type                     count_arrests
   <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.

```r
library(dbplyr)

#map the crime table to tbl
```

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

```
Warning: <BigQueryConnection> uses an old dbplyr interface
i Please install a newer version of the package or contact the maintainer
This warning is displayed once every 8 hours.
```

```
str(crime_table)
```

```
List of 2
 $ unique_key          :List of 2
  ..$ con  :Formal class 'BigQueryConnection' [package "bigrquery"] with 7 slots
  .. .. ..@ project       : chr "bigquery-public-data"
  .. .. ..@ dataset       : chr "chicago_crime"
  .. .. ..@ billing       : chr "refined-circuit-403104"
  .. .. ..@ use_legacy_sql: logi FALSE
  .. .. ..@ page_size     : int 10000
  .. .. ..@ quiet         : logi NA
  .. .. ..@ bigint        : chr "integer"
  ..$ disco: NULL
  ..- attr(*, "class")= chr [1:4] "src_BigQueryConnection" "src_dbi" "src_sql" "src"
 $ case_number         :List of 5
  ..$ x         : 'ident' chr "crime"
  ..$ vars      : chr [1:22] "unique_key" "case_number" "date" "block" ...
  ..$ group_vars: chr(0)
  ..$ order_vars: NULL
  ..$ frame     : NULL
  ..- attr(*, "class")= chr [1:3] "lazy_base_remote_query" "lazy_base_query" "lazy_query"
 - attr(*, "class")= chr [1:5] "tbl_BigQueryConnection" "tbl_dbi" "tbl_sql" "tbl_lazy" ...
```

```
class(crime_table)
```

```
[1] "tbl_BigQueryConnection" "tbl_dbi"                "tbl_sql"
[4] "tbl_lazy"               "tbl"
```

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

```
library(dplyr)

arrest <- crime_table %>%
  filter(year == 2016, district == 11, arrest == TRUE) %>%
  group_by(primary_type) %>%
  summarise(count_arrests = n()) %>%
  arrange(desc(count_arrests)) %>%
  collect() %>%
  head(10) %>%
  print()
```

```
# A tibble: 10 x 2
   primary_type                     count_arrests
   <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
```

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

```
library(dplyr)

arrest2 <- crime_table %>%
  mutate(year=year(date)) %>%
  filter(district == 11, arrest == TRUE) %>%
  group_by(year, primary_type) %>%
  summarise(count_arrests = n()) %>%
  arrange(year) %>%
  collect() %>%
  head(10) %>%
  print()
```

```
`summarise()` has grouped output by "year". You can override using the
```

```
`.groups` argument.

# A tibble: 10 x 3
# Groups:   year [1]
    year primary_type                   count_arrests
   <int> <chr>                                  <int>
 1  2001 ASSAULT                                  322
 2  2001 THEFT                                    419
 3  2001 WEAPONS VIOLATION                        236
 4  2001 BURGLARY                                  42
 5  2001 MOTOR VEHICLE THEFT                      179
 6  2001 SEX OFFENSE                               19
 7  2001 OTHER OFFENSE                            266
 8  2001 ROBBERY                                   97
 9  2001 OFFENSE INVOLVING CHILDREN                44
10  2001 ARSON                                     12
```

Assign the results of the query above to a local R object.

```
arrest_results <- arrest
```

Confirm that you pulled the data to the local environment by displaying the first ten rows of the saved data set.

```
first_ten_rows <- head(arrest_results, n = 10)
print(first_ten_rows)
```

```
# A tibble: 10 x 2
   primary_type                     count_arrests
   <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
```

Close the connection.

```r
dbDisconnect(con)
```