Assignment 4

Due at 11:59pm on November 4.

GitHub repo: https://github.com/Sylviey77/survmeth727-assignment4.git

```
project <- "surv-727-assignment4"
```

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

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

```
<BigQueryConnection>
  Dataset: bigquery-public-data.chicago_crime
  Billing: surv-727-assignment4
```

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.**

```
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 'viviy102077@gmail.com'.

[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.

```sql
SELECT count(primary_type) AS primary_count, count(*) AS overall_count -- counting non-missi
FROM crime
WHERE year = 2016
LIMIT 10;
```

Table 1: 1 records

primary_count	overall_count
269938	269938

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,
  COUNTIF(arrest) AS arrests_2016
FROM crime
WHERE year = 2016
GROUP BY primary_type
ORDER BY arrests_2016 DESC;
```

Table 2: Displaying records 1 - 10

primary_type	arrests_2016
NARCOTICS	13327
BATTERY	10334
THEFT	6522
CRIMINAL TRESPASS	3724
ASSAULT	3494
OTHER OFFENSE	3416
WEAPONS VIOLATION	2510
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?

```
SELECT
  EXTRACT(HOUR FROM date) AS hour_of_day,
  COUNTIF(arrest)         AS arrests_2016
FROM crime
WHERE year = 2016
GROUP BY hour_of_day
ORDER BY arrests_2016 DESC;
```

Table 3: Displaying records 1 - 10

hour_of_day	arrests_2016
19	3843
18	3482
20	3303
21	2962
16	2933
22	2896
11	2893
17	2821
12	2788
14	2775

Answer: Arrests in 2016 peak in the early evening—**19** has the highest count at **3,843**, followed by **18** and **20**.

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

```sql
SELECT
  year,
  COUNTIF(arrest) AS homicide_arrests
FROM crime
WHERE primary_type = 'HOMICIDE'
GROUP BY year
ORDER BY homicide_arrests DESC;
```

Table 4: Displaying records 1 - 10

year	homicide_arrests
2001	431
2002	428
2003	386
2020	356
2022	321
2021	296
2004	294
2016	292
2008	288
2005	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.

```sql
SELECT
  year,
  SAFE_CAST(district AS INT64) AS district,   -- coerce if stored as STRING
  COUNTIF(arrest) AS arrests
FROM crime
WHERE year IN (2015, 2016)
  AND district IS NOT NULL
GROUP BY year, district
ORDER BY arrests DESC, year, district;
```

Table 5: Displaying records 1 - 10

year	district	arrests
2015	11	8975
2016	11	6578
2015	7	5549
2015	15	4514
2015	6	4476
2015	25	4451
2015	4	4326
2015	8	4115
2016	7	3656
2015	10	3628

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.

```
qry_txt <- "
SELECT
  primary_type,
  COUNTIF(arrest) AS arrests_2016_d11
FROM crime
WHERE year = 2016
  AND SAFE_CAST(district AS INT64) = 11   -- robust if district is stored as text
GROUP BY primary_type
ORDER BY arrests_2016_d11 DESC
"
# 1) Create the query object (DBIResult cursor)
qry <- DBI::dbSendQuery(con, qry_txt)

# 2) Fetch all rows into an R data.frame
res_dbi <- DBI::dbFetch(qry)

# 3) Always clear the result
DBI::dbClearResult(qry)

# Peek
head(res_dbi, 10)
```

```
# A tibble: 10 x 2
   primary_type                     arrests_2016_d11
   <chr>                                       <int>
 1 NARCOTICS                                    3634
 2 BATTERY                                       635
 3 PROSTITUTION                                  511
 4 WEAPONS VIOLATION                             303
 5 OTHER OFFENSE                                 255
 6 ASSAULT                                       207
 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
crime_tbl <- dplyr::tbl(con, "crime")

res_dbplyr <- crime_tbl %>%
  dplyr::filter(year == 2016, as.integer(district) == 11) %>%
  dplyr::group_by(primary_type) %>%
  dplyr::summarise(
    arrests_2016_d11 = sum(as.integer(arrest), na.rm = TRUE),
    .groups = "drop"
  ) %>%
  dplyr::arrange(dplyr::desc(arrests_2016_d11))

res_dbplyr
```

```
# Source:     SQL [?? x 2]
# Database:   BigQueryConnection
# Ordered by: dplyr::desc(arrests_2016_d11)
   primary_type                     arrests_2016_d11
   <chr>                                       <int>
 1 NARCOTICS                                    3634
 2 BATTERY                                       635
 3 PROSTITUTION                                  511
 4 WEAPONS VIOLATION                             303
 5 OTHER OFFENSE                                 255
 6 ASSAULT                                       207
 7 CRIMINAL TRESPASS                             205
 8 PUBLIC PEACE VIOLATION                         135
 9 INTERFERENCE WITH PUBLIC OFFICER               119
10 CRIMINAL DAMAGE                                106
# i more rows
```

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

```
res_dbplyr_2016_d11 <- crime_tbl %>%
  dplyr::filter(year == 2016, as.integer(district) == 11) %>%
  dplyr::group_by(primary_type) %>%
  dplyr::summarise(
    arrests_2016_d11 = sum(as.integer(arrest), na.rm = TRUE),  #
    .groups = "drop"
  ) %>%
  dplyr::arrange(dplyr::desc(arrests_2016_d11))

res_dbplyr_2016_d11 %>% head(10)
```

```
# Source:     SQL [?? x 2]
# Database:   BigQueryConnection
# Ordered by: dplyr::desc(arrests_2016_d11)
   primary_type                     arrests_2016_d11
   <chr>                                       <int>
 1 NARCOTICS                                    3634
 2 BATTERY                                       635
 3 PROSTITUTION                                  511
 4 WEAPONS VIOLATION                             303
 5 OTHER OFFENSE                                 255
 6 ASSAULT                                       207
 7 CRIMINAL TRESPASS                             205
 8 PUBLIC PEACE VIOLATION                        135
 9 INTERFERENCE WITH PUBLIC OFFICER              119
10 CRIMINAL DAMAGE                               106
```

```
res_2016_d11_local <- res_dbplyr_2016_d11 %>% dplyr::collect()
head(res_2016_d11_local, 10)
```

```
# A tibble: 10 x 2
   primary_type                     arrests_2016_d11
   <chr>                                       <int>
```

```
 1 NARCOTICS                                3634
 2 BATTERY                                   635
 3 PROSTITUTION                              511
 4 WEAPONS VIOLATION                         303
 5 OTHER OFFENSE                             255
 6 ASSAULT                                   207
 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`.

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

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

```r
crime_tbl <- dplyr::tbl(con, "crime")

res_d11_by_type_year <- crime_tbl %>%
  dplyr::filter(as.integer(district) == 11) %>%
  dplyr::group_by(year, primary_type) %>%
  dplyr::summarise(
    arrests = sum(as.integer(arrest), na.rm = TRUE),
    .groups = "drop"
  ) %>%
  dplyr::arrange(year)

res_d11_by_type_year %>% head(10)
```

```
# Source:     SQL [?? x 3]
# Database:   BigQueryConnection
# Ordered by: year
    year primary_type            arrests
   <int> <chr>                     <int>
 1  2001 CRIM SEXUAL ASSAULT          17
 2  2001 STALKING                      1
 3  2001 CRIMINAL SEXUAL ASSAULT       0
 4  2001 RITUALISM                     0
 5  2001 LIQUOR LAW VIOLATION         49
 6  2001 CRIMINAL DAMAGE             163
 7  2001 INTIMIDATION                   3
 8  2001 OTHER OFFENSE                266
 9  2001 GAMBLING                      71
10  2001 BURGLARY                      42
```

```r
arrests_11_by_type_year <- res_d11_by_type_year %>% dplyr::collect()

head(arrests_11_by_type_year, 10)
```

```
# A tibble: 10 x 3
    year primary_type               arrests
   <int> <chr>                        <int>
 1  2001 RITUALISM                        0
 2  2001 OFFENSE INVOLVING CHILDREN      44
 3  2001 ROBBERY                         97
 4  2001 OTHER OFFENSE                  266
 5  2001 INTIMIDATION                     3
 6  2001 GAMBLING                        71
 7  2001 LIQUOR LAW VIOLATION            49
 8  2001 PROSTITUTION                   424
 9  2001 PUBLIC PEACE VIOLATION          34
10  2001 NARCOTICS                     7979
```

Close the connection.

```r
DBI::dbDisconnect(con)
```