Assignment 4

Due at 11:59pm on November 7.

Akari Oya

This is an individual assignment. Turn in this assignment as an HTML or PDF file to ELMS. Make sure to include the R Markdown or Quarto file that was used to generate it. Include the GitHub link for the repository containing these files.

Github Link = https://github.com/akr-oya/SURVMETH727_Oya.git

Go to https://console.cloud.google.com and make sure you are logged in a non-university Google account. **This may not work on a university G Suite account because of restrictions on those accounts.** Create a new project by navigating to the dropdown menu at the top (it might say "Select a project") and selecting "New Project" in the window that pops up. Name it something useful.

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

```
# Save project ID from Google Cloud
project <- "surv727-database"
```

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

```
# Establish connection to database
con <- dbConnect(
  bigrquery::bigquery(),
  project = "bigquery-public-data",
  dataset = "chicago_crime",
  billing = project
)
con
```

```
<BigQueryConnection>
  Dataset: bigquery-public-data.chicago_crime
  Billing: surv727-database
```

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

```
dbListTables(con) #see names of tables in database
```

```
[1] "crime"
```

```
dbListFields(con, "crime") #see column names
```

```
 [1] "unique_key"          "case_number"          "date"
 [4] "block"               "iucr"                 "primary_type"
 [7] "description"         "location_description" "arrest"
[10] "domestic"            "beat"                 "district"
[13] "ward"                "community_area"       "fbi_code"
[16] "x_coordinate"        "y_coordinate"         "year"
[19] "updated_on"          "latitude"             "longitude"
[22] "location"
```

Using SQL Code

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_
269841

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 count(arrest), primary_type
FROM crime
WHERE year = 2016 AND arrest = TRUE
GROUP BY primary_type
ORDER BY count(arrest) DESC
LIMIT 10;
```

Table 2: Displaying records 1 - 10

f0_	primary_type
13327	NARCOTICS
10332	BATTERY
6522	THEFT
3724	CRIMINAL TRESPASS
3492	ASSAULT
3415	OTHER OFFENSE
2511	WEAPONS VIOLATION
1669	CRIMINAL DAMAGE
1116	PUBLIC PEACE VIOLATION
1097	MOTOR VEHICLE THEFT

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 count(arrest), EXTRACT(HOUR FROM date)
FROM crime
WHERE year = 2016 AND arrest = TRUE
GROUP BY EXTRACT(HOUR FROM date)
LIMIT 15;
```

Table 3: Displaying records 1 - 10

f0_	f1_
3336	5
3750	3
4941	12
5200	11
3609	4
4675	9

f0_	f1_
4735	8
4261	6
4288	1
5306	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 COUNT(arrest) AS arrest_count, year
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

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

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

Table 5: Displaying records 1 - 10

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

Using DBI Package

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.

```
sql <- "SELECT count(arrest) as arrest_count, primary_type
        FROM crime
        WHERE district = 11 AND year = 2016 AND arrest = TRUE
        GROUP BY primary_type
        ORDER BY arrest_count DESC
        LIMIT 10"
```

```
dbGetQuery(con, sql)
```

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

```
 9          119 INTERFERENCE WITH PUBLIC OFFICER
10          106 CRIMINAL DAMAGE
```

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
# Assign data table to object
crime <- dbGetQuery(con, "SELECT * FROM crime") # crime <- tbl(con, "crime") did not work

# Check to see if object is tibble
str(crime)
```

```
tibble [7,925,652 x 22] (S3: tbl_df/tbl/data.frame)
 $ unique_key          : int [1:7925652] 11582399 3941611 6822713 10930557 5948406 1924060 1
 $ case_number         : chr [1:7925652] "JC136206" "HL306996" "HR229223" "JA246103" ...
 $ date                : POSIXct[1:7925652], format: "2019-02-01 10:32:00" "2005-04-18 07:00
 $ block               : chr [1:7925652] "050XX W MADISON ST" "100XX S WOODLAWN AVE" "050XX
 $ iucr                : chr [1:7925652] "1821" "1350" "0281" "1156" ...
 $ primary_type        : chr [1:7925652] "NARCOTICS" "CRIMINAL TRESPASS" "CRIM SEXUAL ASSAUL
 $ description         : chr [1:7925652] "MANU/DEL:CANNABIS 10GM OR LESS" "TO STATE SUP LAND
 $ location_description: chr [1:7925652] "SIDEWALK" "COLLEGE/UNIVERSITY RESIDENCE HALL" "APA
 $ arrest              : logi [1:7925652] TRUE TRUE FALSE FALSE TRUE TRUE ...
 $ domestic            : logi [1:7925652] FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ beat                : int [1:7925652] 1533 511 1533 1533 1533 1022 1533 1022 1023 1021 ..
 $ district            : int [1:7925652] 15 5 15 15 15 10 15 10 10 10 ...
 $ ward                : int [1:7925652] 28 8 28 28 28 NA 29 24 28 24 ...
 $ community_area      : int [1:7925652] 25 50 25 25 25 NA 25 29 29 29 ...
 $ fbi_code            : chr [1:7925652] "18" "26" "02" "11" ...
 $ x_coordinate        : num [1:7925652] 1142721 1186531 1142775 1142867 1144408 ...
 $ y_coordinate        : num [1:7925652] 1899552 1838828 1898530 1898863 1899062 ...
 $ year                : int [1:7925652] 2019 2005 2009 2017 2007 2001 2015 2017 2004 2012 .
 $ updated_on          : POSIXct[1:7925652], format: "2019-02-08 04:14:17" "2018-02-28 03:56
 $ latitude            : num [1:7925652] 41.9 41.7 41.9 41.9 41.9 ...
 $ longitude           : num [1:7925652] -87.8 -87.6 -87.8 -87.8 -87.7 ...
 $ location            : chr [1:7925652] "(41.880419254, -87.751407336)" "(41.7128602, -87.59
```

```r
class(crime)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

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

```
crime %>%
  select(arrest, primary_type, district, year) %>%
  filter(district == 11 & year == 2016 & arrest == TRUE) %>%
  group_by(primary_type) %>%
  count()
```

```
# A tibble: 27 x 2
# Groups:   primary_type [27]
   primary_type                          n
   <chr>                             <int>
 1 ARSON                                 2
 2 ASSAULT                             206
 3 BATTERY                             635
 4 BURGLARY                             22
 5 CONCEALED CARRY LICENSE VIOLATION     2
 6 CRIM SEXUAL ASSAULT                  10
 7 CRIMINAL DAMAGE                     106
 8 CRIMINAL SEXUAL ASSAULT               3
 9 CRIMINAL TRESPASS                   205
10 DECEPTIVE PRACTICE                   63
# i 17 more rows
```

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.

```
# Assign filtered data into another object
sql_data <-
  crime %>%
  select(year, arrest, district, primary_type) %>%
  filter(district == 11 & arrest == TRUE) %>%
  group_by(year, primary_type) %>%
  summarise(n = n()) %>%
  collect()
```

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

```
# Show first ten rows of the saved data set
head(sql_data, n = 10)
```

```
# A tibble: 10 x 3
# Groups:   year [1]
    year primary_type            n
   <int> <chr>               <int>
 1  2001 ARSON                  12
 2  2001 ASSAULT               322
 3  2001 BATTERY               962
 4  2001 BURGLARY               42
 5  2001 CRIM SEXUAL ASSAULT    17
 6  2001 CRIMINAL DAMAGE       163
 7  2001 CRIMINAL TRESPASS     389
 8  2001 DECEPTIVE PRACTICE     84
 9  2001 GAMBLING               71
10  2001 HOMICIDE               48
```

Close the connection.

```
dbDisconnect(con)
```