R sqldf Package

I. "sqldf" Package

1) Install "sqldf" Package

```
devtools::install_github("ggrothendieck/sqldf")
```

```
Downloading GitHub repo ggrothendieck/sqldf@HEAD

bit     (4.0.4 -> 4.0.5 ) [CRAN]
plogr   (NA    -> 0.2.0 ) [CRAN]
proto   (NA    -> 1.0.0 ) [CRAN]
chron   (NA    -> 2.3-61) [CRAN]
RSQLite (NA    -> 2.3.1 ) [CRAN]
gsubfn  (NA    -> 0.7   ) [CRAN]
Installing 6 packages: bit, plogr, proto, chron, RSQLite, gsubfn

Installing packages into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

── R CMD build ─────────────────────────────────────────────────────────────────
* checking for file '/tmp/RtmpxRfhp9/remotes9ec93c3eb/ggrothendieck-sqldf-ba33807/DESCRIPTION' ... OK
* preparing 'sqldf':
* checking DESCRIPTION meta-information ... OK
* checking for LF line-endings in source and make files and shell scripts
* checking for empty or unneeded directories
* building 'sqldf_0.4-11.tar.gz'

Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)
```

2) Installed Packages

```
library()
```

3) Import "sqldf" Package

```
library(sqldf)
```

```
Loading required package: gsubfn

Loading required package: proto

Warning message:
"no DISPLAY variable so Tk is not available"
Loading required package: RSQLite
```

II. Load Data

- read.csv( )

```
url = 'https://raw.githubusercontent.com/rusita-ai/pyData/master/PII.csv'

TB <- read.csv(url)
```

- str( )

```
str(TB)
```

```
'data.frame':   17 obs. of  8 variables:
 $ Name     : chr  "송태섭" "최유정" "이한나" "김소혜" ...
 $ Gender   : chr  "남자" "여자" "여자" "여자" ...
 $ Age      : int  21 23 20 23 24 24 22 22 23 22 ...
 $ Grade    : int  3 1 1 3 4 2 4 2 3 2 ...
 $ Picture  : chr  "무" "유" "무" "무" ...
 $ BloodType: chr  "B" "A" "A" "O" ...
 $ Height   : num  179 177 168 176 176 ...
 $ Weight   : num  63.9 54.9 50.2 53.5 79.8 61.7 69.8 52.7 68.5 52.3 ...
```

- head( )

```
head(TB, 5)
```

III. SQL 수행

1) "SELECT * FROM Table_Name"

```
sqldf("SELECT Height FROM TB")
```

- Import "dplyr" Package

```
library(dplyr)
```

```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

```
sqldf("SELECT Gender, Height FROM TB") %>% head(3)
```

```
sqldf("SELECT Name, Height FROM TB") %>% head(3)
```

```
sqldf("SELECT Gender, Height, BloodType FROM TB") %>% head(3)
```

```
sqldf("SELECT * FROM TB") %>% head(3)
```

2) "WHERE" 조건구문

```
sqldf("SELECT * FROM TB WHERE Height > 175")
```

(1) "AND" 연산

```
sqldf("SELECT * FROM TB WHERE Height > 175 AND Weight < 75")
```

```
sqldf("SELECT * FROM TB WHERE Height > 175 AND Grade = 3")
```

(2) "OR" 연산

```
sqldf("SELECT * FROM TB WHERE Height > 175 OR Grade = 3")
```

```
sqldf("SELECT * FROM TB WHERE Height > 175 OR Age = 24")
```

3) 명목형 연산

```
sqldf("SELECT * FROM TB WHERE Name = '강백호'")
```

```
sqldf("SELECT * FROM TB WHERE BloodType = 'B'")
```

```
sqldf("SELECT * FROM TB WHERE Height > 175 AND BloodType = 'B'")
```

4) "IN" 연산자

```
sqldf("SELECT * FROM TB WHERE Grade IN ('2', '4')")
```

```
sqldf("SELECT * FROM TB WHERE BloodType IN ('A', 'B', 'O')")
```

5) "LIKE" 연산자

- 특정 문자 시작, 끝, 포함하는 값을 추출

```
sqldf("SELECT * FROM TB WHERE Weight LIKE '5%'")
```

```
sqldf("SELECT * FROM TB WHERE Weight LIKE '7%'")
```

```
sqldf("SELECT * FROM TB WHERE Weight LIKE '%9'")
sqldf("SELECT * FROM TB WHERE Weight LIKE '%2'")
```

```
sqldf("SELECT * FROM TB WHERE Name LIKE '김%'")
```

```
sqldf("SELECT * FROM TB WHERE Name LIKE '%정'")
```

```
sqldf("SELECT * FROM TB WHERE Name LIKE '%소%'")
```

6) "GROUP BY" 연산자

- 중복값을 제외하고 1개의 고유값만 출력

```
sqldf("SELECT Grade FROM TB GROUP BY Grade")
```

7) "SUM( )" 연산

```
sqldf("SELECT Grade, SUM(Age), SUM(Height), SUM(Weight)
          FROM TB  GROUP BY Grade")
```

8) "AVG( )" 연산

```
sqldf("SELECT Grade, AVG(Age), AVG(Height), AVG(Weight)
          FROM TB  GROUP BY Grade")
```

9) "HAVING" 연산자

```
sqldf("SELECT Grade, AVG(Age), AVG(Height), AVG(Weight)
          FROM TB
          GROUP BY Grade
          HAVING AVG(Height) > 170")
```

10) "ORDER BY" 연산자

- 오름차순 : **ASC** , 내림차순 : **DESC**

```
sqldf("SELECT * FROM TB WHERE Height > 170")
```

- Height 기준

```
sqldf("SELECT * FROM TB WHERE Height > 170 ORDER BY Height ASC")
```

```
sqldf("SELECT * FROM TB WHERE Height > 170 ORDER BY Height DESC")
```

- Weight 기준

```
sqldf("SELECT * FROM TB WHERE Height > 170 ORDER BY Weight ASC")
```

```
sqldf("SELECT * FROM TB WHERE Height > 170 ORDER BY Weight DESC")
```

11) "AS" 연산자

- 출력 Column Name 변경

```
sqldf("SELECT SUM(Height) AS SUM_Height FROM TB")
```

```
sqldf("SELECT AVG(Height) AS AVG_Height, AVG(Weight) AS AVG_Weight
        FROM TB")
```

# # #

End Of Document

# # #