

# Homework13a

Steve Simon

This file was created on 2020-07-24 and last modified on 2020-07-25.

Note: this solution uses R and SQLite. An alternate solution using SAS and Oracle is also available.

Note: Some of the names used in this code are arbitrary and you can choose whatever names you want. To emphasize which names can be modified at your discretion, I am using names of famous statisticians.

The statistician being honored in this code is Hirotugu Akaike.

Use the longitudinal\_example\_db. It is available as on the Canvas website or you can find it on the Insights platform.

1. Do an inner join of baseline\_table and one\_year\_table. Display the first ten rows of data only.

```
library(sqldf)

## Loading required package: gsubfn
## Loading required package: proto
## Loading required package: RSQLite

akaike <- dbConnect(SQLite(),
  dbname="../../data/longitudinal_example_db.sqlite")
hirotugu1 <- dbGetQuery(conn=akaike, "
  select b.id, b.pk1, o.pk5
  from baseline_table as b
  join one_year_table as o
  on b.id=o.id
  limit 10
")

hirotugu1

##      id  pk1      pk5
## 1  104 16.00 15.33333
## 2  108 16.50 23.25000
## 3  112  9.25  6.25000
## 4  113 42.50 51.25000
## 5  114 24.25 25.25000
## 6  126 21.00 15.25000
## 7  130 21.75  1.00000
## 8  131 14.50  2.50000
## 9  135 40.50 28.75000
## 10 137 11.75 13.50000

dbDisconnect(conn=akaike)
```

2. Explain why id 104 included in this inner join, but not the inner join shown in the video?

3. Count the number of records after an inner join of `baseline_table` and `year_one_table`. Compare this to the number of records in the `year_one_table`.

```
library(sqldf)
akaike <- dbConnect(SQLite(),
  dbname="../data/longitudinal_example_db.sqlite")
hirotugu3a <- dbGetQuery(conn=akaike, "
  select count(b.id) as n
    from baseline_table as b
   join one_year_table as o
   on b.id=o.id
")
```

```
hirotugu3a
```

```
##      n
## 1 301
```

```
hirotugu3b <- dbGetQuery(conn=akaike, "
  select count(o.id) as n
    from one_year_table as o
")
```

```
hirotugu3b
```

```
##      n
## 1 301
```

```
dbDisconnect(conn=akaike)
```

4. Compute the average pk score at baseline, the average score at one year, and the average change score.

```
library(sqldf)
akaike <- dbConnect(SQLite(),
  dbname="../data/longitudinal_example_db.sqlite")
hirotugu4 <- dbGetQuery(conn=akaike, "
  select
    avg(b.pk1) as pk1_avg,
    avg(o.pk5) as pk5_avg,
    avg(b.pk1)-avg(o.pk5) as change_score
  from baseline_table as b
  join one_year_table as o
  on b.id=o.id
")
```

```
hirotugu4
```

```
##      pk1_avg  pk5_avg change_score
## 1 25.56894 19.08245      6.486489
```

```
dbDisconnect(conn=akaike)
```

5. Find and list the two labels in `migraine_table` that do not correspond to any codes in `demog_table`.

```
library(sqldf)
akaike <- dbConnect(SQLite(),
  dbname="../data/longitudinal_example_db.sqlite")
hirotugu5 <- dbGetQuery(conn=akaike, "
```

```

select
  m.migraine_code, m.migraine_label
  from migraine_table as m
  left join demog_table as d
  on m.migraine_code=d.migraine
  where d.migraine is null
")

```

hirotugu5

```

##   migraine_code migraine_label
## 1             98           Other
## 2             99           Unknown

```

```
dbDisconnect(conn=akaike)
```

6. Show that there are no unmatched labels or unmatched codes for group\_table.

```

library(sqldf)
akaike <- dbConnect(SQLite(),
  dbname="../data/longitudinal_example_db.sqlite")
hirotugu6 <- dbGetQuery(conn=akaike, "
  select
    count(g.group_code) as n
    from group_table as g
    left join demog_table as d
    on g.group_code=d.grp
    where d.grp is null
")

```

hirotugu6

```

##   n
## 1 0

```

```
dbDisconnect(conn=akaike)
```