Selecting All Observations When Any Observation Is of Interest

Christopher Bost
What does that mean?

- Some data have more than one obs/person
- We want to select all obs for a person
  - If at least one obs for that person meets criteria
- Examples?
What's in it for you?

- Review how to select obs with the DATA step
- Review how to select obs with PROC SQL
- Useful techniques to have in your SAS "toolkit"
Sample data set: COURSES

<table>
<thead>
<tr>
<th>studentid</th>
<th>course</th>
<th>ap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BIOL101</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>BIOL102</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CHEM102</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>CALC101</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>STAT101</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>PSYC201</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>HIST102</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>PHYS101</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CHEM101</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>CALC102</td>
<td>1</td>
</tr>
</tbody>
</table>
### Desired data set

<table>
<thead>
<tr>
<th>studentid</th>
<th>course</th>
<th>ap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>BIOL102</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CHEM102</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>PSYC201</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>HIST102</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>PHYS101</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CHEM101</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>CALC102</td>
<td>1</td>
</tr>
</tbody>
</table>
Select observations with DATA step

1. Subset observations of interest
2. Keep one observation per person
3. Match-merge with original data set
1. Subset observations of interest

data ap;
set courses;
where ap=1;
run;

Data set AP

<table>
<thead>
<tr>
<th>studentid</th>
<th>course</th>
<th>ap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>BIOL102</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CHEM102</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>PSYC201</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>PHYS101</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CALC102</td>
<td>1</td>
</tr>
</tbody>
</table>
2. Keep one observation per person

proc sort data=ap out=ap_sort;
by studentid;
run;

data ap2;
set ap_sort;
by studentid;
if first.studentid;
keep studentid;
run;

Data set AP2

<table>
<thead>
<tr>
<th>studentid</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>
3. Match-merge observations

```plaintext
proc sort data=courses out=courses_sort;
by studentid;
run;

data anyap;
merge courses_sort ap2 (in=inap2);
by studentid;
if inap2;
run;
```
Desired data set: ANYAP

<table>
<thead>
<tr>
<th>studentid</th>
<th>course</th>
<th>ap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>BIOL102</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CHEM102</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>PSYC201</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>HIST102</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>PHYS101</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CHEM101</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>CALC102</td>
<td>1</td>
</tr>
</tbody>
</table>
data ap;
set courses;
where ap=1;
run;

proc sort data=ap
    out=ap_sort;
by studentid;
run;

data ap2;
set ap_sort;
by studentid;
if first.studentid;
keep studentid;
run;

proc sort data=courses
    out=courses_sort;
by studentid;
run;

data anyap;
merge courses_sort
    ap2 (in=inap2);
by studentid;
if inap2;
run;
Pros and cons

- It works
- Three DATA steps
- Two PROC SORT steps
Select observations with PROC SQL

1. Use a subquery
2. Use GROUP BY and HAVING clauses
# Terminology

<table>
<thead>
<tr>
<th>DATA Step</th>
<th>PROC SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>Column</td>
</tr>
<tr>
<td>Observation</td>
<td>Row</td>
</tr>
<tr>
<td>SAS data set</td>
<td>Table</td>
</tr>
</tbody>
</table>
SQL clauses: required order

 PROC  SQL  ;  
  
  1  SELECT  
  
  2  FROM  
  
  3  WHERE  
  
  4  GROUP  BY  
  
  5  HAVING  
  
  6  ORDER  BY  ;  
  
  QUIT  ;  

 *starts procedure*

 *selects variables*

 *opens data sets*

 *restricts observations*

 *groups observations*

 *restricts groups*

 *sorts results*

 *ends procedure*
Use a subquery

```sql
proc sql;
create table anyap2 as
select *
from courses
where studentid in (select distinct studentid
                        from courses
                        where ap=1)
order by studentid;
quit;
```
<table>
<thead>
<tr>
<th>studentid</th>
<th>course</th>
<th>ap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>BIOL102</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CHEM102</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>PSYC201</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>PHYS101</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>HIST102</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>CALC102</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CHEM101</td>
<td>0</td>
</tr>
</tbody>
</table>
Pros and cons

- Single step
- Two passes through data set
  - One for the subquery
  - One for the outer query
SQL clauses: order of execution

1 FROM opens data sets
2 WHERE restricts observations
3 GROUP BY groups observations
4 HAVING restricts groups
5 SELECT selects variables
6 ORDER BY sorts results
Use GROUP BY and HAVING clauses

```sql
proc sql;
create table anyap3 as
select *
from courses
  group by studentid
  having sum(ap=1) > 0
order by studentid;
quit;
```
Desired data set: ANYAP3

<table>
<thead>
<tr>
<th>studentid</th>
<th>course</th>
<th>ap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>CHEM102</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>BIOL102</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>HIST102</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>PSYC2101</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>PHYS101</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CALC102</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CHEM101</td>
<td>0</td>
</tr>
</tbody>
</table>
Pros and cons

- Single step
- Flexibility
  - Any condition(s) in parentheses after SUM
- NOTE: The query requires remerging summary statistics back with the original data.
Conclusion

- Use either the DATA step or PROC SQL
- PROC SQL requires significantly less coding
- PROC SQL is not necessarily more efficient
  - Test on your own data
- Use on any data with groups of observations
Contact information

Comments and questions are valued and encouraged.

Christopher J. Bost
MDRC
16 East 34th Street
New York, NY 10016
(212) 340-8613
christopher.bost@mdrc.org
chrisbost@gmail.com
Alternate DATA step method

```sas
proc sort data=courses(where=(ap=1) keep=studentid ap) out=lookup(keep=studentid) nodupkey;
by studentid;
run;

proc sort data=courses out=courses_sort;
by studentid;
run;

data anyap;
merge courses_sort lookup(in=inlookup);
by studentid;
if inlookup;
run;
```
Alternate PROC SQL method

proc sql;
create table anyap4 as
select courses.*
from courses inner join
  (select distinct studentid as studentid2
   from courses
   where ap=1)
on studentid=studentid2
order by studentid;
quit;