Next Presentation: Searching for Variable Values with CAT Functions

Presenter: Mike Zdeb

Mike is an assistant professor in the epidemiology & biostatistics department at the U@Albany School of Public Health in Rensselaer, NY. He's been a SAS user for 20+ years and has presented papers at SUGI, SAS Global Forums, NESUG, and numerous local user groups. Mike has written a SAS Press book, *Mapping Made Easy Using SAS*, and has been a reviewer for a number of SAS Press books.
## NEW FUNCTIONS IN V9.1

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT</td>
<td>Concatenates <em>character</em> strings without removing leading or trailing blanks</td>
</tr>
<tr>
<td>CATS</td>
<td>Concatenates <em>character</em> strings and removes leading and trailing blanks</td>
</tr>
<tr>
<td>CATT</td>
<td>Concatenates <em>character</em> strings and removes trailing blanks only</td>
</tr>
<tr>
<td>CATX</td>
<td>Concatenates <em>character</em> strings, removes leading and trailing blanks, and inserts separators</td>
</tr>
</tbody>
</table>
Searching for Variable Values with CAT Functions

NEW FUNCTION IN V9.2

**CATQ** concatenates *character or numeric* values by using a delimiter to separate items and by adding quotation marks to strings that contain the delimiter.

The **CATX** function is similar to the **CATQ** function except that **CATX** does not add quotation marks included "modifiers" that change the behavior of the function (no modifiers for other CAT functions)
results of the CAT, CATS, CATT, and CATX functions are usually equivalent to results that are produced by certain combinations of the concatenation operator (||) and the TRIM and LEFT functions
## Searching for Variable Values with CAT Functions

| Function           | Equivalent Code (with conCATenation operator ||) |
|--------------------|--------------------------------------------------|
| CAT(OF X1-X4)      | X1 || X2 || X3 || X4                           |
| CATS(OF X1-X4)     | TRIM(LEFT(X1)) || TRIM(LEFT(X2)) || TRIM(LEFT(X3)) || TRIM(LEFT(X4)) |
| CATS(OF X1-X4)     | STRIP(X1) || STRIP(X2) || STRIP(X3) || STRIP(X4)          |
| CATT(OF X1-X4)     | TRIM(X1) || TRIM(X2) || TRIM(X3) || TRIM(X4)          |
| CATX(SP, X1-X4)    | TRIM(LEFT(X1)) || SP || TRIM(LEFT(X2)) || SP || TRIM(LEFT(X3)) || SP || TRIM(LEFT(X4)) |
| CATX(SP, X1-X4)    | STRIP(X1) || SP || STRIP(X2) || SP || STRIP(X3) || SP || STRIP(X4) |
Searching for Variable Values with CAT Functions

- default length for the CAT, CATS, CATT, and CATX functions is different from the length that is obtained when you use the concatenation operator

```plaintext
data names;
input last_name : $25. first_name : $15. @@;

* old way;
full_name_old_way = trim(last_name) || ', ' || first_name;

* new way;
full_name_new_way = catx(', ',last_name,first_name);

datalines;
Zdeb Mike Washington George Squirrel Rocket(J);
```

Searching for Variable Values with CAT Functions

<table>
<thead>
<tr>
<th>last_name</th>
<th>first_name</th>
<th>full_name_old_way</th>
<th>full_name_new_way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zdeb</td>
<td>Mike</td>
<td>Zdeb, Mike</td>
<td>Zdeb, Mike</td>
</tr>
<tr>
<td>Washington</td>
<td>George</td>
<td>Washington, George</td>
<td>Washington, George</td>
</tr>
<tr>
<td>Squirrel</td>
<td>Rocket(J)</td>
<td>Squirrel, Rocket(J)</td>
<td>Squirrel, Rocket(J)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alphabetic List of Variables and Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
Searching for Variable Values with CAT Functions

- NOT JUST FOR CONCATENATION
- ANOTHER USE ... SEARCH FOR VARIABLE VALUES

For example, given a data set named ANSWERS

Create a new data set containing observations where at least one answer had a value of "Y"
Searching for Variable Values with CAT Functions

- CONVENTIONAL ARRAY AND A LOOP

  ```plaintext
  data yes;
  set answers;
  array a(10);
  do j=1 to 10 until (a(j) eq 'Y');
  end;
  if j lt 11;
  drop j;
  run;
  ```

- CAT AND FIND FUNCTIONS

  ```plaintext
  data yes;
  set answers;
  if find(cat(of a1-a10),'Y');
  run;
  ```
Searching for Variable Values with CAT Functions

- UP TO NOW ... NO ONE HAS ASKED

```plaintext
data yes;
set answers;
if whichc('Y', of a:);
run;
```

- WHY NOT USE WHICHC ... NO REASON IN THIS EXAMPLE, BUT ...

```plaintext
data bad;
set answers;
if verify(cat(of a:),'YN');
run;
```
NEW TASK

create a new data set containing observations where at least TWO answers had a value of "Y"
Searching for Variable Values with CAT Functions

### CONVENTIONAL ARRAY AND A LOOP

```plaintext
data yes;
set answers;
array a(10);
do j=1 to 10 until (found_y eq 2);
   found_y = sum(found_y, a(j) eq 'Y');
end;
if found_y eq 2;
drop j found_y;
run;
```

### CAT AND COUNTC FUNCTIONS

```plaintext
data yes;
set answers;
if countc(cat(of a:),'Y') ge 2
run;
```
NEW TASK

add a new variable to each observation in data set ANSWERS

the variable will contain the number of answers that are "Y"
Searching for Variable Values with CAT Functions

* array and a loop;
data yes;
set answers;
array a(10);
* use _n_ as loop iterator (no DROP);
do _n_ = 1 to 10;
  found_y = sum(found_y, (a(_n_) eq 'Y'));
end;
run;

* cat and countc;
data yes;
set answers;
found_y = countc(cat(of a:), 'Y'));
run;
Search for Variable Values with CAT Functions

- **NEW DATA SET WITH NUMERIC ANSWERS**

- **SIMILAR TASK**

create a new data set containing observations where at least one answer had a value of 5

<table>
<thead>
<tr>
<th></th>
<th>id</th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>a5</th>
<th>a6</th>
<th>a7</th>
<th>a8</th>
<th>a9</th>
<th>a10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1234</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A2345</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>A3456</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A4567</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>A5678</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>
Searching for Variable Values with CAT Functions

- USE CAT AND FIND
- NO PROBLEM WITH NUMERIC DATA (SEARCH FOR '5')

SAS log ...

243  * cat and find;
244  data five;
245  set answers;
246  if find(cat(of a:),'5');
247  run;

NOTE: There were 5 observations read from the data set WORK.ANSWERS.
NOTE: The data set WORK.FIVE has 4 observations and 11 variables.
Searching for Variable Values with CAT Functions

- YES ... YOU COULD HAVE USED ... if max (of a:) eq 5;

- AND, YES ... YOU COULD HAVE USED WHICHN ...

```plaintext
data five;
set answers;
if whichn(5, of a:);
run;
```
Searching for Variable Values with CAT Functions

- **BUT WHAT ABOUT ...**

```plaintext
data bad;
set answers;
if verify(cat(of a:),'012345');
run;
```
Searching for Variable Values with CAT Functions

A TEST ...

```sas
data five;
set answers;
answers = cat(of a:);
if find(answers,'5');
run;
```

WHAT TYPE OF VARIABLE IS ANSWERS ... CHAR OR NUM (WHY?)
Searching for Variable Values with CAT Functions

HINT (THE LOG) ...

```sas
1497    data five;
1498    set answers;
1500    answers = cat(of a:);
1501    if find(answers,'5');
1502    run;

NOTE: Character values have been converted to numeric values at the places given by: (Line):(Column).
1500:11
NOTE: Numeric values have been converted to character values at the places given by: (Line):(Column).
1501:9
NOTE: There were 5 observations read from the data set WORK.ANSWERS.
NOTE: The data set WORK.FIVE has 4 observations and 12 variables.
NOTE: DATA statement used (Total process time):
     real time       0.00 seconds
     cpu time       0.00 seconds
```
Searching for Variable Values with CAT Functions

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>a1</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>a2</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>a3</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>a4</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>a5</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>a6</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>a7</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>a8</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>a9</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>a10</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>answers</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>id</td>
<td>Char</td>
<td>1</td>
</tr>
</tbody>
</table>

Alphabetic List of Variables and Attributes

<table>
<thead>
<tr>
<th>id</th>
<th>a1</th>
<th>a2</th>
<th>a3</th>
<th>a4</th>
<th>a5</th>
<th>a6</th>
<th>a7</th>
<th>a8</th>
<th>a9</th>
<th>a10</th>
<th>answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5533211111</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>555</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5555555555</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1010105553</td>
</tr>
</tbody>
</table>
Searching for Variable Values with CAT Functions

- DIGRESSION ... QUESTION FROM SAS-L

Is there anyway to check first three digits of numeric value?

- HOW ABOUT THE CAT FUNCTION

```sas
data x;
input abc @@;
if cat(abc) eq : '111';
datalines;
111222 123456 11111111 987654 111;
```

- THIS WORKS
Searching for Variable Values with CAT Functions

THE LOG ... NO ERRORS ...

2     data x;
3     input abc @@;
4     if cat(abc) eq : '111';
5     datalines;

NOTE: SAS went to a new line when INPUT statement reached past the end of a line.
NOTE: The data set WORK.X has 3 observations and 1 variables.
NOTE: DATA statement used (Total process time):
real time 0.59 seconds
cpu time 0.04 seconds
Searching for Variable Values with CAT Functions

- ANOTHER TEST ... THIS DOES NOT WORK (WHY?)

```plaintext
data x;
input abc @@;
if cat(abc) eq : '111';
datalines;
111222  123456  11111111  987654  111 11  1;
```

- HINT ... WHAT DOES eq : REALLY MEAN

- FIX ... if cat(abc,'x') eq '111';
MORE COMPLEXITY ... SEARCH FOR DIAGNOSIS CODES

each observation has up to five, three character diagnosis codes and the code 250 indicates diabetes ... find patient records with diabetes

| 025  | 022 |
| 682  | 401  | 244  | 493  |
| 592  | 401  | 493  |
| 428  | 493  | 780  | V43  | 250  |
| 250  |
| 414  | V45  | 401  | 250  |
Searching for Variable Values with CAT Functions

- INCORRECT ... WHY?

```plaintext
data dia_incorrect;
set diag3;
if find(cat(of dx1-dx5),'250') ne 0; run;
```

- WHAT IS THE RESULT OF THE CAT FUNCTION?

- LOOK AT THE FIRST RECORD ... THERE IS A 250 BUT NEITHER DIAGNOSIS IS 250 (A COMBINATION OF 025 AND 022)

<table>
<thead>
<tr>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>025022</td>
</tr>
<tr>
<td>682401244493</td>
</tr>
<tr>
<td>592401493</td>
</tr>
<tr>
<td>428493780V43250</td>
</tr>
<tr>
<td>250</td>
</tr>
<tr>
<td>414V45401250</td>
</tr>
</tbody>
</table>
Searching for Variable Values with CAT Functions

- CORRECT ... USE `CATX`

```plaintext
data dia_correct;
set diag3;
if find(catx('*',of dx1-dx5),'250') ne 0;
run;
```

- WHAT IS THE RESULT OF THE CATX FUNCTION

- LOOK AT THE FIRST RECORD ... THERE IS NO 250

```
025*022
682*401*244*493
592*401*493
428*493*780*V43*250
250
414*V45*401*250
```
CATX MAKES IT EASY TO SEARCH FOR VARIOUS DIAGNOSES

250 indicates diabetes, 493 indicates asthma ... find patient records with diabetes and/or asthma

data dia_ast;
set diag3;
dia = (find(catx('*',of dx1-dx5),'250') ne 0);
ast = (find(catx('*',of dx1-dx5),'493') ne 0);
run;

CODE ADDS DUMMY VARIABLES: 1, PRESENT; 0, NOT PRESENT
Searching for Variable Values with CAT Functions

In case you are already asking ...

data dia_ast;
set diag3;
dia = whichc('250', of dx:) ne 0;
ast = whichc('493', of dx:) ne 0;
run;
Searching for Variable Values with CAT Functions

MORE COMPLEXITY ...
FIVE CHARACTER
DIAGNOSIS CODES,
ONLY THE FIRST THREE
CHARACTERS USED TO
FIND DISORDERS

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25001</td>
<td>V180</td>
<td>5680</td>
<td>78039</td>
<td>6250</td>
<td>49390</td>
</tr>
<tr>
<td>4270</td>
<td>4111</td>
<td>4019</td>
<td>36250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>486</td>
<td>7990</td>
<td>25082</td>
<td>41400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78659</td>
<td>25000</td>
<td>49320</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40392</td>
<td>4660</td>
<td>2449</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34839</td>
<td>2765</td>
<td>40493</td>
<td>4280</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ONCE AGAIN ... FIND DIABETES (250) AND/OR ASTHMA (493)
Searching for Variable Values with CAT Functions

- COULD USE AN ARRAY PLUS A LOOP

```sas
data dia_ast;
set diag5;
array dx(5);
da = 0;
ast = 0;
do j=1 to 5;
   if dx(j) eq : '250' then dia = 1;
   if dx(j) eq : '493' then ast = 1;
end;
drop j;
run;
```

- EQ:
LOOKS FOR VALUES THAT START WITH THE SPECIFIED STRING
Searching for Variable Values with CAT Functions

- OR, USE CATX FUNCTION ... INCORRECT ... WHY?

```sas
data dia_ast_incorrect;
  set diag5;
  dia = (find(catx('*',of dx1-dx5), '*250') gt 0);
  ast = (find(catx('*',of dx1-dx5), '*493') gt 0);
run;
```

- WHAT IS THE RESULT OF THE CATX FUNCTION?

- LOOK AT THE FIRST RECORD ... STARTS WITH 250

```
25001*V180
5680*78039*6250*49390*53081
4270*4111*4019*36250
486*7990*25082*41400
78659*25000*49320
49392*4660*2449
34839*2765*40493*4280
```
MODIFIED CATX FUNCTION

```sas
data dia_ast_correct;
set diag5;
dia = (find(catx('*','*','of dx1-dx5'),'*250') gt 0);
ast = (find(catx('*','*','of dx1-dx5'),'*493') gt 0);
run;
```

WHAT IS THE RESULT OF THE CATX FUNCTION

```
**25001*V180
**5680*78039*6250*49390*53081
**4270*4111*4019*36250
**486*7990*25082*41400
**78659*25000*49320
**49392*4660*2449
**34839*2765*40493*4280
```
"REAL LIFE" EXAMPLE

data set with 2.5 million observations ... each observation has an admit, principal, and up to fourteen other diagnoses

identify observations with either: diabetes (250); asthma (493); heart attack (410); female breast cancer (174); influenza (487)
DONE THREE DIFFERENT WAYS

* use an array and a loop;
data array_loop;
set temp;
array dx(16) adx pdx odx1-odx14;
dia = 0;
ast = 0;
ami = 0;
brc = 0;
flu = 0;
do j=1 to 16 until (dx(j) eq ' ');
   if dx(j) eq : '410' then ami = 1;
   if dx(j) eq : '174' then brc = 1;
   if dx(j) eq : '487' then flu = 1;
end;
drop j;
Searching for Variable Values with CAT Functions

* CAT and FIND with a new variable
  (and a LENGTH statement);

data cat_find_new;
set temp;
length string $100;
string = catx('**',adx,pdx,of odx:);
dia = (find(string,'*250') gt 0);
ast = (find(string,'*493') gt 0);
ami = (find(string,'*410') gt 0);
brc = (find(string,'*174') gt 0);
flu = (find(string,'*487') gt 0);
drop string;
run;
Searching for Variable Values with CAT Functions

* CAT and FIND with repeated concatenation;

```plaintext
data cat_find_repeat;
set temp;

dia = (find(catx('*','*',adx, pdx,of odx:),'*250') gt 0);
ast = (find(catx('*','*',adx, pdx,of odx:),'*493') gt 0);
ami = (find(catx('*','*',adx, pdx,of odx:),'*410') gt 0);
bro = (find(catx('*','*',adx, pdx,of odx:),'*174') gt 0);
flu = (find(catx('*','*',adx, pdx,of odx:),'*487') gt 0);
run;
```
Searching for Variable Values with CAT Functions

- EACH METHOD TRIED 25 TIMES
- DATA SET (NAMED TEMP) IN MEMORY USING SASFILE
  
  ```
  sasfile temp load;
  ```

- TIMES ARE IN SECONDS

<table>
<thead>
<tr>
<th>METHOD</th>
<th>ELAPSED TIME</th>
<th>CPU TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRAY AND LOOP</td>
<td>21.1</td>
<td>17.4</td>
</tr>
<tr>
<td>CAT AND FIND WITH A NEW VARIABLE</td>
<td>19.9</td>
<td>16.4</td>
</tr>
<tr>
<td>CAT AND FIND WITH A NEW VARIABLE (AND A LENGTH STATEMENT)</td>
<td>18.4</td>
<td>13.3</td>
</tr>
<tr>
<td>CAT AND FIND WITH REPEATED CONCATENATION</td>
<td>18.6</td>
<td>14.3</td>
</tr>
</tbody>
</table>
Searching for Variable Values with CAT Functions

WHAT IF SELECTION RULES ARE COMPLEX

Traumatic brain injury includes ICD-9 codes 800-801.9, 803-804.9, 850-854.1, 950.1-950.3, 959.01 and 995.55.

```
proc format;
value $tbi
'800'-'80199', '803'-'80499' , '850'-'85419',
'9501'-'95039', '95901' , '99555' = '1'
other = '0';
run;
```
Searching for Variable Values with CAT Functions

CONCATENATE FORMATTED VALUES OF VARIABLES

```sas
filename nosee dummy;

data tbi;
set temp;
file nosee;
put (adx pdx odx1-odx14) ($tbi.) @;
if find(_file_,"1") then output;
put;
run;

catt(put(adx,$tbi.), put(pdx,$tbi.) ...
    put(odx14,$tbi.));
```
SEARCHING RULES NOT COMPLEX ... ANOTHER APPROACH FOR LOOKING AT THE FIRST THREE CHARACTERS IN FIVE CHARACTER DX CODES ... CONCATENATE FIRST THREE CHARACTERS

```plaintext
filename nosee dummy;
data tbi;
set temp;
file nosee;
put (adx pdx odx1-odx14) ($3. +1) @;
if find(_file_,'250') then output;
put;
run;
```
Searching for Variable Values with CAT Functions

LOOK FOR MULTIPLE DIAGNOSES ... SAVE PATTERN

```plaintext
proc format;
value $dia_ast
   '250' - '25099' = 'D' '493' - '49399' = 'A'
   other = 'x';
run;

data dia_ast;
length pattern $16;
set temp;
file nosee;
put (adx pdx odx1-odx14) ($dia_ast.) @;
pattern = _file_; 
if findc(_file_,'DA') then output;
put; 
run;
```
## Searching for Variable Values with CAT Functions

<table>
<thead>
<tr>
<th>pattern</th>
<th>adx</th>
<th>pdx</th>
<th>odx1</th>
<th>odx2</th>
<th>odx3</th>
<th>odx4</th>
<th>odx5</th>
<th>odx6</th>
<th>odx7</th>
<th>odx8</th>
<th>odx9</th>
<th>odx10</th>
<th>odx11</th>
<th>odx12</th>
<th>odx13</th>
<th>odx14</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>5990</td>
<td>78079</td>
<td>25000</td>
<td>41400</td>
<td>2967</td>
<td>78057</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxxDxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>49121</td>
<td>41519</td>
<td>78079</td>
<td>7850</td>
<td>41400</td>
<td>25000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>82123</td>
<td>V4589</td>
<td>7140</td>
<td>25000</td>
<td>2727</td>
<td>2721</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxxDxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>496</td>
<td>4280</td>
<td>78079</td>
<td>7850</td>
<td>25000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>40391</td>
<td>V451</td>
<td>25001</td>
<td>78079</td>
<td>2449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxxxDxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>486</td>
<td>5119</td>
<td>78079</td>
<td>41012</td>
<td>4019</td>
<td>2720</td>
<td>25000</td>
<td>4439</td>
<td>7804</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>4280</td>
<td>78079</td>
<td>25080</td>
<td>41400</td>
<td>5939</td>
<td>4019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxDxxxxxxxx</td>
<td>49121</td>
<td>49121</td>
<td>7990</td>
<td>25000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxxxDxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>78079</td>
<td>70710</td>
<td>4581</td>
<td>V1251</td>
<td>4019</td>
<td>43310</td>
<td>25000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>78079</td>
<td>V126</td>
<td>7806</td>
<td>25000</td>
<td>2727</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxxDxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>25001</td>
<td>9986</td>
<td>78079</td>
<td>V1089</td>
<td>V1251</td>
<td>4019</td>
<td>2809</td>
<td>2449</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxxDxxxxx</td>
<td>49320</td>
<td>49320</td>
<td>2761</td>
<td>2765</td>
<td>4580</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>71536</td>
<td>V4365</td>
<td>41400</td>
<td>4019</td>
<td>25000</td>
<td>42789</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>8082</td>
<td>25001</td>
<td>V5789</td>
<td>V1582</td>
<td>73300</td>
<td>2720</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>78079</td>
<td>V1089</td>
<td>V4589</td>
<td>25000</td>
<td>4019</td>
<td>2720</td>
<td>2449</td>
<td>53081</td>
<td>2809</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>78079</td>
<td>92300</td>
<td>25002</td>
<td>4019</td>
<td>41400</td>
<td>43310</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>71535</td>
<td>V4364</td>
<td>25001</td>
<td>436</td>
<td>42731</td>
<td>V1259</td>
<td>2720</td>
<td>53081</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>80709</td>
<td>V1005</td>
<td>V4589</td>
<td>496</td>
<td>41400</td>
<td>25000</td>
<td>4111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxxxxxDxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>78079</td>
<td>V4972</td>
<td>V4975</td>
<td>V4971</td>
<td>40391</td>
<td>V451</td>
<td>25071</td>
<td>44381</td>
<td>74789</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>78079</td>
<td>2900</td>
<td>41400</td>
<td>25060</td>
<td>25050</td>
<td>3371</td>
<td>3572</td>
<td>36201</td>
<td>5363</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>78079</td>
<td>4139</td>
<td>4280</td>
<td>2851</td>
<td>25000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>71535</td>
<td>V4364</td>
<td>25000</td>
<td>2859</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xxxxDxxxxxx</td>
<td>V5789</td>
<td>V5789</td>
<td>4019</td>
<td>25000</td>
<td>49390</td>
<td>78057</td>
<td>4169</td>
<td>45981</td>
<td>4540</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Searching for Variable Values with CAT Functions

- COMPARE

```plaintext
data cat_find_repeat;
set temp;
dia = (find(catx('**','**',adx, pdx,of odx:),'*250') gt 0);
ast = (find(catx('**','**',adx, pdx,of odx:),'*493') gt 0);
ami = (find(catx('**','**',adx, pdx,of odx:),'*410') gt 0);
brc = (find(catx('**','**',adx, pdx,of odx:),'*174') gt 0);
flu = (find(catx('**','**',adx, pdx,of odx:),'*487') gt 0);
run;
```

```plaintext
data use_file_; 
set temp; 
file nosaic; 
put (adx pdx odx1-odx14) ($3. +1) @; 
dia = (find(_file_, '250') gt 0); 
ast = (find(_file_, '493') gt 0); 
ami = (find(_file_, '410') gt 0); 
brc = (find(_file_, '174') gt 0); 
flu = (find(_file_, '487') gt 0); 
put; 
run;
```
LAST DIGRESSION ... FROM SAS COMMUNITY

if I concatenate several data sets, is there any way to find out what data set an observation came from
Searching for Variable Values with CAT Functions

- **VERSION 9.2+ ... CAN USE INDSNAME**

```sql
data all;
set one two three
indsname=temp;
dsn = temp;
run;
```

- **VERSION 9.13**

```sql
data all;
set one (in=_1) two (in=_2) three(in=_3);
from = find(catt(of _:),'1');
run;
```
Searching for Variable Values with CAT Functions

CONTACT INFORMATION ...

msz03@albany.edu

www.albany.edu/~msz03

www.sascommunity.org/wiki/User:Msz03

518 402 6479