What Does a SAS User Need to Know about Numeric Precision

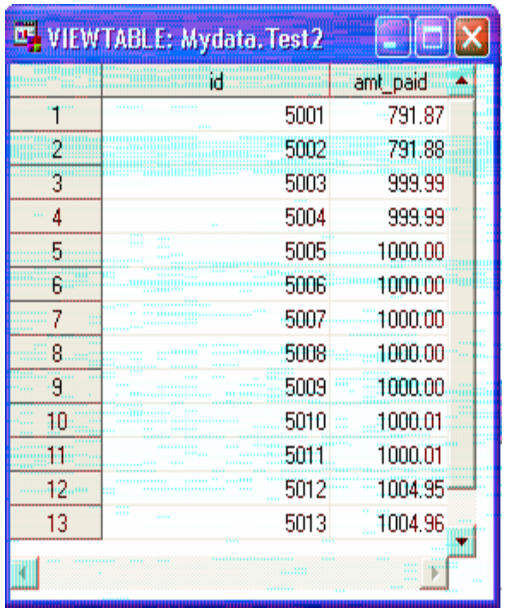
Jason (Jianduan) Liang
Consultant, CGI
Email: jianduan.liang@cgi.health.gov.ab.ca

Agenda

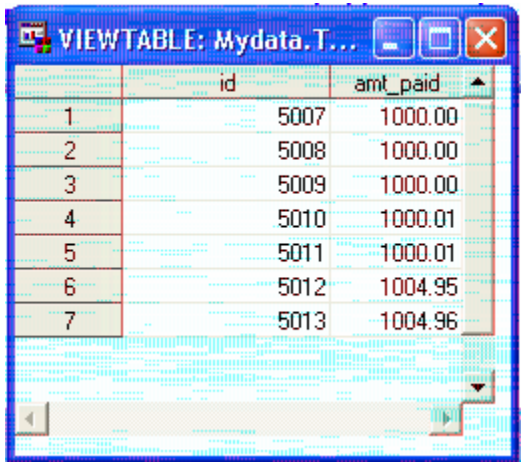
- Why numeric precision?
- Example issues
- Where the issues came from?
- Solutions

Example Issues

```
data mydata.test3;
   set mydata.test2;
   if amt_paid ge 1000;
run;
```



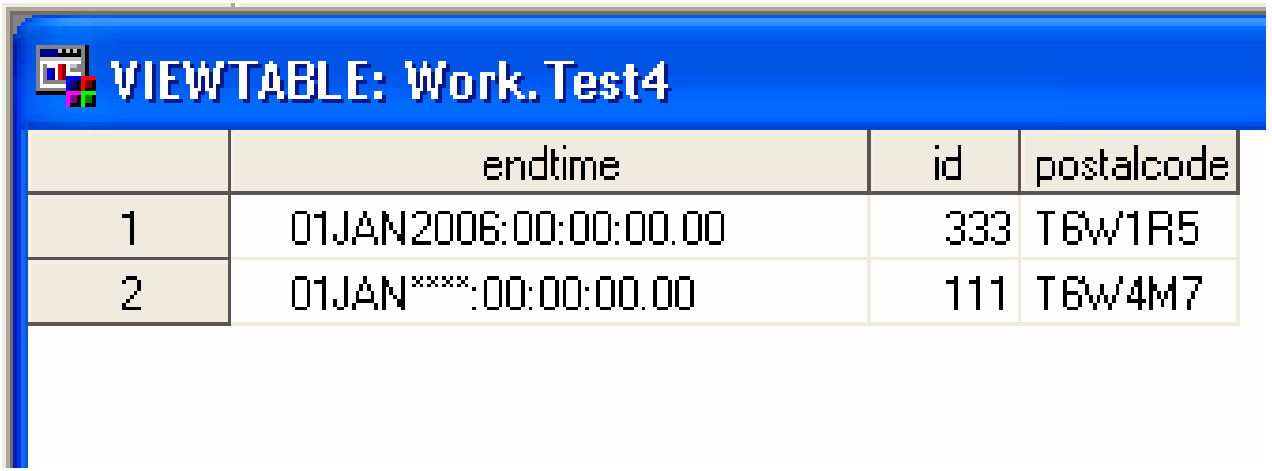
	id	amt_paid
1	5001	791.87
2	5002	791.88
3	5003	999.99
4	5004	999.99
5	5005	1000.00
6	5006	1000.00
7	5007	1000.00
8	5008	1000.00
9	5009	1000.00
10	5010	1000.01
11	5011	1000.01
12	5012	1004.95
13	5013	1004.96



	id	amt_paid
1	5007	1000.00
2	5008	1000.00
3	5009	1000.00
4	5010	1000.01
5	5011	1000.01
6	5012	1004.95
7	5013	1004.96

Example issues

```
data test4;
    format endtime datetime25.2;
    input id postalcode $6. endtime datetime25.6;
   datalines;
333         T6W1R5        31DEC2005:23:59:59.999999
111         T6W4M7        31DEC9999:23:59:59.999999
;
run;
```



VIEWTABLE: Work.Test4

	endtime	id	postalcode
1	01JAN2006:00:00:00.00	333	T6W1R5
2	01JAN****:00:00:00.00	111	T6W4M7

Reasons

- People are customized to decimal arithmetic.
- Computers use binary arithmetic with finite precision.

  $184=128+32+16+8=2^7+2^5+2^4+2^3$

  10111000
- Some numbers (real numbers) do not have exact binary representations.

Example number without exact binary representation

```
data a;
    point_three=0.3;
    three_times_point_one=3*0.1;
    difference=point_three-
three_times_point_one;
proc print noobs;
run;
```

point_ three	three_ times_ point_one	difference
0.3	0.3	-5.5511E-17

Floating Point Representation

- **SAS uses Floating Point Representation to store numeric values**

- **Floating Point Representation is an implementation of scientific notation**

Decimal value 978 = .978 x 10^{3}

Value=mantissa x base ^ exponent

  - mantissa – determines the precision by which the number is stored.
  - exponent – determines the magnitude of the number.
  - base – determines number system

Floating Point Representation

- SAS uses 8 bytes to store a numeric value unless you specify differently.
- 64 bit layout

SEEEEEEE EEEEMMMM MMMMMMMM

MMMMMMMM MMMMMMMM MMMMMMMM

MMMMMMMM MMMMMMMM

S – sign E – exponent M – mantissa

BIAS (1023)– SAS uses it for the exponent so that both positive and negative exponents can be stored without an additional sign bit.

Floating Point Representation-example

255.75

128+64+32+16+8+4+2+1+1/2+1/4

$$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0+2^{-1}+2^{-2}$$

255 = 11111111 .75 = 11

1111 1111.11 -> 1.111 1111 11

mantissa 1111 1111 1 => 1111 1111 1000

F F 8

exponent 7 +1023=1030=>0100 0000 0110

4 0 6

Foating point representation for 255.75 is 406FF80000000000

Floating Point Representation example

In decimal system

1/3 0.33333…

3*1/3=0.99999…

In SAS

0.1 can not be exactly represented, therefore, 3*0.1 is not exactly 0.3.

What should we do with the issues?

- **Be aware of the issues**
- **Use comparison functions such as ROUND**

```
data mydata.test5;
    set mydata.test2;
    if int(round(amt_paid,.01)*100) ge 100000;
run;
```