Combinatorial analysis & calculations using SAS functions – `fact()`, `perm()` and `comb()`

fortuneWalla

August 16, 2015

Created on Friday, March 7th, 2014 at 5:56 am

SAS provides several kinds of functions for doing combinatorial analysis and calculations. Three basic ones will be demonstrated here. `fact()`,`perm()` and `comb()`. All three functions return a missing value for the arguments they cannot compute.

`fact(n)` is the function for calculating the factorial `n!` of any non-negative number `n`.

```
275  data _null_;
276  a=fact(-3);
277  b=fact(0);
278  c=fact(9);
279  d=fact(170);*170! is the max calculable by this particular computer.;
280  e=fact(171);
281  f=fact(1000);
282  put _all_;
283  run;
```

The output is given by:

```
NOTE: Invalid argument to function FACT at line 276 column 3.
NOTE: Invalid argument to function FACT at line 280 column 3.
NOTE: Invalid argument to function FACT at line 281 column 3.
a=. b=1 c=362880 d=7.257416E306 e=. f=. _ERROR_=1 _N_=1
a=. b=1 c=362880 d=7.257416E306 e=. f=. _ERROR_=1 _N_=1
NOTE: Mathematical operations could not be performed at the following places. The results of the
      operations have been set to missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      1 at 276:3   1 at 280:3   1 at 281:3
NOTE: DATA statement used (Total process time):
```

Similarly for permutation of `n` objects taken `r` at a time (where `n>r`), we have the `perm(n,r)`. A single argument in the `perm()` function will calculate the factorial of the argument.

```
384  data _null_;
385  a=perm(3,3);
386  b=perm(2,5);
387  c=perm(5,2);
388  d=perm(5,4);
389  e=perm(1660,170);
390  f=perm(4);
391  put _all_;
392  run;
```

The output is given by

```
NOTE: Argument 2 to function PERM at line 386 column 3 is invalid.
NOTE: Invalid argument to function PERM at line 389 column 3.
a=6 b=. c=20 d=120 e=. f=24 _ERROR_=1 _N_=1
a=6 b=. c=20 d=120 e=. f=24 _ERROR_=1 _N_=1
NOTE: Mathematical operations could not be performed at the following places. The results of the
      operations have been set to missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      1 at 386:3   1 at 389:3
NOTE: DATA statement used (Total process time):
```

Similarly for combination of `n` objects taken `r` at a time (where `n>r`), we have the `comb(n,r)`. The `comb()` function requires two arguments.

```
433  data _null_;
434  a=comb(3,3);
435  b=comb(2,5);
436  c=comb(5,2);
437  d=comb(5,4);
438  e=comb(9960,170);
439  f=comb(1,0);
440  put _all_;
441  run;
```

The output is given by:

```
NOTE: Argument 2 to function COMB at line 435 column 3 is invalid.
NOTE: Invalid argument to function COMB at line 438 column 3.
a=1 b=. c=10 d=5 e=. f=1 _ERROR_=1 _N_=1
a=1 b=. c=10 d=5 e=. f=1 _ERROR_=1 _N_=1
NOTE: Mathematical operations could not be performed at the following places.
      The results of the operations have been set to missing values.
      Each place is given by: (Number of times) at (Line):(Column).
      1 at 435:3   1 at 438:3
```

We have the logarithmic (natural) counterparts of the above three functions i.e. `lfact()`, `lperm()` and `lcomb()` The output is given below.

```
452  data _null_;
453  a=lfact(10);
454  b=lperm(10,5);
455  c=lcomb(10,5);
456  put _all_;
457  run;

a=15.104412573 b=10.31692083 c=5.5294290875 _ERROR_=0 _N_=1
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```