# Filling in the Blanks

Notes on Expressions: PDQ-Explore 06/20/03

The entries in the PDQ-Explore setups are often simply the names of individual items such as sex, race, or state. However, items are just the simplest form of the general expressions that can be entered. An expression typically is made up of one or more item names linked by arithmetic and/or logical operations: plus (+), minus (-), divide (/), equal (=), and (&), or (|), etc. The full list of PDQ-Operations is given in the table below along with the level of precedence for each.

Expressions may be simple or complex. A simple expression may be used to collapse age to a more manageable number of categories, for example:

```
age/10
```

or wage and salary income to $1,000 intervals:

```
income1/1000
```

Complex expressions may be used to allow a characteristic of a married person to be related to a characteristic of the spouse or a characteristic of a child to be related to a characteristic of a father or mother. See the recodes under the heading "PDQ Custom Items" for the 1990 5% PUMS in the PDQ-Explore Workspace window to see a variety of examples of expressions along with the assignment of identifying names to the new categories generated by the expressions.

Operations with higher levels of precedence are executed before lower levels unless parentheses are used to control the order of execution. When parentheses are used, execution occurs within the inner-most parentheses first.

Consider the following example where the logical AND will be executed before the logical OR if not for the parentheses:

```
state=26 & (age<18 | age>=65)
```

The OR within the parentheses is executed first to select persons less than 18 years of age or older than 65. The result (TRUE or FALSE) is then combined with the result of state=26 (Michigan) through the AND. The result will be TRUE if the person is from Michigan and age is either less than 18 or greater than 65. Otherwise, the result will be FALSE.

If the parentheses were not present, all persons older than 65 will be included as TRUE in the result along with persons under 18 who resided in Michigan--probably not the intended result.

| Precedence Level | Operator | Name | Example/Comment |
|---|---|---|---|
| 9 | X:a..b | range | age:15..44 |
| 8 | unary + | plus | sex=+1 (never needed) |
| 8 | unary - | minus | income4<=-1000 |
| 7 | * | multiply | 73*income1/100 |

| | | | |
|---|---|---|---|
| 7 | / | divide | rhhinc/persons |
| 7 | % | modulo | subsample%10 |
| 6 | + | add | income1+income2 |
| 6 | - | subtract | rhhinc-rearning |
| 5 | < | less than | age<65 |
| 5 | > | greater than | age>64 |
| 5 | <= | less than or equal | age<=65 |
| 5 | >= | greater than or equal | age>=65 |
| 4 | = or == | equal | age=23 |
| 4 | != or <> | not equal | income!=0 |
| 3 | & or && | and | race=2 & looking=1 |
| 2 | ^ | exclusive or | bit-wise--use with caution |
| 1 | \| or \|\| | or | age<18 \| age>=65 |

Logical TRUE evaluates as a numeric 1; logical FALSE evaluates as 0 in numeric expressions. All non-zero numeric values are TRUE in logical expressions.

Use parentheses freely to control the order of execution of operations, especially if the effect of precedence is not obvious.

The range operator specifies the range of values that are to be included in a query and displayed in the results. For example, the tabulation of age:15..44 as the row axis will give counts for each age from 15 through 44. The range occasionally needs to be specified in this manner for recodes and more complex expressions where the PDQ-Explore software cannot reliably determine the possible range of results.

Four intrinsic functions, $sum, $min, $max, and $pick, are available that loop through the records in a hierarchy and return the sum, minimum, maximum, or selected values, respectively, of the argument, which is typically an arithmetic or logical expression. In the simplest case, sum returns the count of the number of persons in the housing unit with a given trait. For example: sum(age<18) returns the number of persons under age 18 in the housing unit; min(age) returns the age of the youngest person in the housing unit; and max(yearsch) returns the yearsch value for the most highly educated person in the housing unit.

The pick function has the structure pick(expression,item). The function returns the value of the item for the first record within a lower hierarchy for which the expression is true. For example for the 1990 PUMS, pick(relat1=1,age) will return the age of the spouse of the head, if present.

A fifth intrinsic function, abs, returns the absolute value of the expression within the parentheses: abs(expression).

Additional arithmetic functions perform numeric transformations on item values:

| Function | Result |
|---|---|
| $ceil(exp) | Round up |
| $floor(exp) | Round down |
| $log(exp) | Log base e |
| $log10(exp) | Log base 10 |

```
    $pow(exp1,exp2)   Exp1 raised to exp2 power
    $rint(exp)        Round to nearest integer
    $sqrt(exp)        Square root
    $sin(exp)         Sine
    $cos(exp)         Cosine
    $tan(exp)         Tangent
    $asin(exp)        Arcsine
    $acos(exp)        Arccosine
    $atan(exp)        Arctangent
```

Selections are necessary when results are not defined for specific values, such as 0 or negative numbers in the case of logarithms.

The results of these functions are truncated to integer values when used in PDQ-Explore expressions. Thus, scaling will often be needed to obtain meaningful results: 100*$sin(exp), for example.

A case function is available to map a sequence of expressions to results:

```
    case(exp1,rslt1,exp2,rslt2,exp3,rslt3,...,default result)
```

For example, `case(race=1,1,race=2,2,(race>=3 & race<=299),3,4)`

Examples of expressions:

```
select:    sex=1 & age>=15 & age<=49
select:    (age>=15 & age<=65) & (occup=84 | occup=85)
row:       age:15..25
row:       age>=15 & age<=25
describe:  log(income1)
```

Note that the two row examples do not yield the same results:

    age:15..25 will display one row for each single year of age in the 15-25 range.

    age>=15 & age<=25 will display False and True rows for those cases outside and those within the 15-25 year range, respectively.