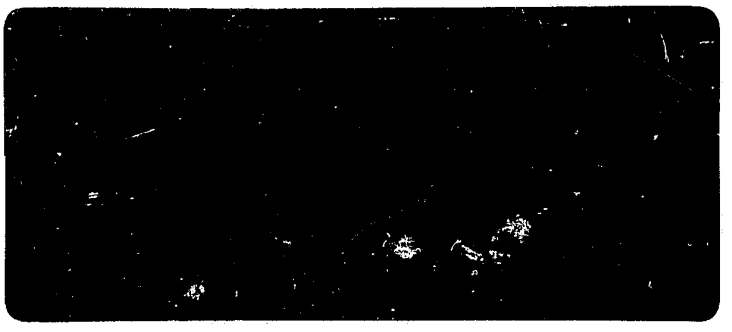
BSSR



BUREAU OF SOCIAL SCIENCE RESEARCH, INC.

WASHINGTON, D. C.

INTERACTIVE TABLE GENERATOR

ITG COMMANDS

Produced Under Grant 78-SS-AX-0028

for

U.S. Department of Justice
Law Enforcement Assistance Administration
National Criminal Justice Information and
Statistics Service

Bureau of Social Science Research
1990 M Street N.W.
Washington, D.C. 20036

June 27, 1979

TITLE
TITLE Command
Function:
Creates a title for a table.

Syntax:

TITLE title-text.

Example

TITLE Criminal Justice Expenditures / 1971 Through 1975

A slash,"/", in the text causes a new line in the title.
Two slashes together print as a single slash in the title.
Two slashes separated by a blank cause lines of the title to be double spaced.

HEADING
HEADING or COLUMN Command

Function:

Determines the structure of the columns of a table. Together the STUB statement and the HEADING Statement define the tabulations to be performed and specify the form of the table. COLUMN is a synonym for HEADING. A maximum of 5 columns can be defined.

Syntax:

COLUMN|HEADING control-var (value-list) BY|THEN control-variable (value-list) ...

control-var
The name of an old or new control variable, or the new control variable, or the word TOTAL. The use of TOTAL is discussed below.

value-list
A list of control variable values or categories to be included in the table. Single values are separated by commas. A range of values is indicated by two values separated by a colon.

BY The nesting operator, causes all possible combinations of the categories of the two variables to be displayed. AGE BY INCOME produces columns containing

AGE 1 INCOME 1, AGE 1 INCOME 2, AGE 2 INCOME 1, and AGE 2 INCOME 2.

THEN
The concatenation operator, causes all of the categories of the first control variable to be displayed, then all categories of the second variable.

Examples:

AGE THEN INCOME yields AGE 1, AGE 2, INCOME 1, INCOME 2.

HEADING SECTOR(1:3) THEN TOTAL

HEA RACE(1,3) BY SEX
In the second example, four columns will be produced: SECTOR 1, SECTOR 2, and SECTOR 3, followed by the TOTAL of the three sectors.

BY is said to take precedence over THEN which means nesting will be done first and concatenation last. For example, AGE BY EDUC THEN INCOME produces six columns containing:

AGE 1 EDUC 1, AGE 1 EDUC 2, AGE 2 EDUC 1, AGE 2 EDUC 2, INCOME 1, INCOME 2.

Precedence can be altered with parentheses. AGE BY (EDUC

THEN INCOME) is equivalent to the expression AGE BY EDUC THEN AGE BY INCOME. Assuming two categories for each control variable, the result is a table with eight columns: the nesting of AGE BY EDUC, followed by the nesting of AGE BY INCOME.

The keyword TOTAL can be used with the concatenation operator, THEN, to generate totals across columns. For example, AGE THEN TOTAL gives each AGE category followed by the total over all AGE categories. (TOTAL THEN INCOME) BY AGE yields six columns:
TOTAL INCOME AGE 1, TOTAL INCOME AGE 2, INCOME 1 AGE 1, INCOME 1 AGE 2, INCOME 2 AGE 1, INCOME 2 AGE 2.

STUB
STUB or ROW Command

Function:

Determines the structure of the stubs or rows of a table and the content of the table. Together the STUB statement and the HEADING Statement define the tabulations to be performed and specify the form of the table. ROW is a synonym for STUB.

As defined here, a stub is a group of rows of a table generated by the expression "obs-var BY control-var". Stubs are separated in the command by semicolons.

Syntax:

STUB|ROW obs-var BY control-var (value-list) ...
; obs-var BY control-var (value-list) ... ...

obs-var
The name of an observation variable, i.e., the variable to be aggregated. Observation variables constitute the content of the cells of a table.

control-var
The name of an old or new control variable, or the word TOTAL. The use of TOTAL is discussed below.

value-list
A list of control variable values or categories to be included in the table. Single values are separated by commas. A range of values is indicated by two values separated by a colon.

BY The nesting operator, causes all possible combinations of the categories of the variables to be displayed. AGE BY INCOME produces columns containing

AGE 1 INCOME 1, AGE 1 INCOME 2, AGE 2 INCOME 1, and AGE 2 INCOME 2.

Examples

STUB EXPEND BY SECTOR; TOTAL; FTE

STU *_CRIMES BY OFFENSE(1:3) BY STATE(1,4,8)

The first example will produce a table displaying expenditures broken down by sector of expenditures, the total of expenditures, followed by the (total) number of full-time equivalent employees.

In the second example, the number of crimes for offense categories 1 through 3, for states 1, 4, and 8 will be displayed. The order of display will be OFFENSE 1 STATE 1, OFFENSE 1 STATE 4, OFFENSE 1 STATE 8, OFFENSE 2 STATE 1, etc.

GO
GO Command

Function:

Signals the end of the table specification phase of IT6 and causes the table requested to be tabulated and displayed at the terminal.

Syntax:

GO

STOP
STOP Command

Function:

Terminates ITG and returns to MTS.

Syntax:
STOP

HELP
HELP Command

Function:

The HELP Command can be used at any time to obtain information about the purpose of ITG, ITG Commands, and data available to ITG. The synonym for HELP is "?". "HELP" or "?" may be entered at any time there is any doubt about what to do next.

Syntax:

HELP help-expression

Where help-expression is one of the following:

<null>
If nothing is entered after "HELP", then a default help message is displayed. This may be an explanation of ITG, the next command expected, or a list of items for which HELP is available.

ITG
A description of the purpose and operation of the Interactive Table Generator.

COMMANDS
Displays a list and brief description of ITG commands.

command
Information about the function and use of the ITG command named.

FILES file-name
If no file name is given, a list and brief description of all available data files is displayed. If a file name is specified, a more detailed description of the sources and content of the file is given.

SUBFILE subfile-name
Description of a subfile within a data file, including the observation and control variable names and the control variable values.

DATA variable-name
If no variable is named, then a description of the selected data file is displayed. A more complete description of an observation variable including applicable control variable names and levels is given when an observation variable is named. Further information about a control variable is displayed by specifying a control variable name. Before the HELP DATA command can be used, a data file must be selected. See the USE command.

TERMS term
The HELP TERM command lists a glossary of terms with abbreviations and brief definitions. A more complete definition of a term can be obtained by specifying its name or abbreviation in the command.

REPLACE
REPLACE or RESET Command

Function:

Change any previcusly entered command. This is an alternative to re-entering a lengthy command such as STUB.
Syntax:

REPLACE|RESET command n | n | ALL

If no command is specified the default is the last table specification command entered. If a command is given then that command is selected for correction. Valid commands are: TITLE, HEADING, STUB, COMPUTE, POST COMPUTE, DEFINE, LABEL and SELECT.

FILE may also be specified; however, selecting a new file destroys all other table specifications. Specifying ALL destroys all table specifications, including the choice of data file.

"n" is the nth occurance or the nth expression of the selected command. The commands, COMPUTE, POST COMPUTE. DEFINE, LABEL and SELECT can be used more than once for each table, and STUB and HEADING can have more than one expression. The value of n is displayed as each command is entered. The value of n is displayed as each command is entered, or can be obtained by using the LIST command prior to the REPLACE command. If n is not given, ITG will display all occurances or expressions of a command and the sequence number, n. ITG will then ask the user to enter n; and prompt for the corrected entry. The user may then enter the new information or press carriage return to delete the entry.

LIST
LIST Command
Function:

Display existing table specifications.

Syntax:

LIST command-name

command-name
The name of the command specifications to be displayed.

Examples:
LIST STUB
LIS TITLE

If the specifications for only one command are needed, the command name can be included. Valid commands are USE, TITLE, HEADING, STUB, COMPUTE, POST COMPUTE, DEFINE, LABEL and SELECT.

Each occurance of the COMPUTE, POST COMPUTE, DEFINE, LABEL and SELECT commands and each expression in the HEADING and STUB commands is sequence numbered. This number can be used with the REPLACE command to correct an entry.

COMPUTE
COMPUTE Command

Function:

Create a new observation variable.

Syntax:

COMPUTE variable-name : arithmetic expression.

variable-name
The name of the new observation variable.

arithmetic-expression
A signed or unsigned numeric constant or observation variable.

An algebraic expression composed of numeric constants, observation variables and numeric operators, "+","-","*" and "/", and parentheses.

Examples:

HOMICIDE = MURDER + MANSLATR
NEWVAR = (OLDVAR /2.5 + 46 * -1.4) / 100

A numeric constant is a string of numbers, 0 through 9, and optionally, a decimal point or a sign, "+" or "-". Commas may not be used. The maximum number of digits is nine if the constant is a whole number. If the constant contains a decimal point the maximum number of digits is seven.

Any observation variable from the data file or defined in a previous COMPUTE statement can be used.

The numeric operators are:

+	addition	*	multiplication
-	subtraction	/	division

Calculation of an arithmetic expression proceeds from left to right. If an expression contains more than one operator the precedence order (order of calculation) is multiplication or division first, then addition and subtraction. The order of calculation can be modified by placing parts of the expression in parentheses. Expressions inside parentheses are done before calculations outside parentheses. Parenthesis may be nested. The following example illustrates the order in which computation is done.

```
X = (-4*(2+3))+(2*-1.5)
  = (-4*5)+(2*-1.5)
  = -20+(2*-1.5)
  = -20+(-3)
  = -23.
```

POST COMPUTE
POST COMPUTE Command
Function:

Create a new observation variable using aggregated values of observation variables. These computations are done after the values in the table have been aggregated.
Syntax:
POST COMPUTE new-obs-var: arithmetic-expression

variable-name
The name of the new observation variable.

arithmetic-expression
A signed or unsigned numeric constant or observation variable.

An algebraic expression composed of numeric constants, observation variables and numeric operators, "+","-","*" and "/", and parentheses.

RATE (obs-var-1, n, obs-var-2) which calculates the rate of observation variable 1 per "n" of observation variable 2, e.g., rate of arrest per 1,000 population.

Examples:

POST COMPUTE AVGINCOM : INCOME / PERSONS.

POST C ARST_RAT : RATE(ARRESTS,1000,USPOP)
A numeric constant is a string of numbers, 0 through 9, and optionally, a decimal point and/or a sign, "+" or "-". Commas may not be used. The maximum number of digits is nine if the constant is a whole number. If the constant contains a decimal point the maximum number of digits is seven.

Any observation variable from the data file or defined in a previous COMPUTE statement can be used.
The numeric operators are:

+	addition	*	multiplication
-	subtraction	/	division

Calculation of an arithmetic expression proceeds from left to right. If an expression contains more than one operator the precedence order (order of calculation) is multiplication or division first, then addition and subtraction. The order of calculation can be modified by placing parts of the expression in parentheses. Expressions inside parentheses are done before calculations outside parentheses. Parenthesis may be nested. The following example illustrates the order in which computation is done.

X = (-4*(2+3))+(2*-1.5)

```
= (-4*5)+(2*-1.5)
= -20+(2*-1.5)
= -20+-3
= -23.
```

DEFINE
DEFINE or COLLAPSE Command

Function:

Create a new control variable by grouping or ordering an existing control or observation variable.

Syntax:

DEFINE|COLLAPSE control-variable-name ON old-variable-name; value IF condition-entry; value IF condition-entry; ....

control-variable-name
The name of the newly defined control variable.

old-variable name
The name of an existing control or observation variable whose values determine the values of the new control variable.

value
The values of the new control variable. Values must be zero or positive integers, but need not be in sequence.

condition-entry
The values of the old variable. The condition-entry can take the following forms:

Old variable value.

Range of old variable values. expressed as "low value : high value ".

Relation symbol followed by an old variable value. Relation symbols are:

=	equal	¬=	not equal
<	less than	¬<	not less than
>	greater than	¬>	not greater than
<=	less than or equal to	>=	greater than or equal to

The word "ALL" meaning all values of the old variable.
The word "OTHER" meaning any values not specified in another condition entry.

Examples:

DEF INCOME ON WAGES; 1 IF <5000; 2 IF 5000:9999; 3 IF >=10000

COLLAPSE REGION ON STATE; 0 IF ALL; 1 IF 1; 2 IF 3; 1 IF 2; 1 IF 6; 2 IF 4;

In the first example, the new control variable INCOME is 1 if the old variable WAGES is less than 5000, 2 if WAGES is 5000 through 9999, and 3 if Wages is greater or equal to 10000.

The second example illustrates a powerful grouping ability. The values of the new control variable can be defined from over lapping values of the old variable. In this case all states are grouped under the value zero. Then states 1, 2, and 6 are grouped under the value 1 and states 3 and 4 are grouped under the value 2.

Values for the new control variable can be omitted. If the new value is omitted, the keyword IF can also be omitted. If no new values appear then ITb will generate values beginning with 1.

Also, if several values of the old variable are to be grouped under one value of the new control variable, then the new value need be specified only once at the beginning of the group of the old values. The grouping requested in the second example could be specified as:

COLLAPSE REGION ON STATE; 0 IF ALL; 1 IF 1,2,6; 2 IF 3:5;

A series of old values are separated by commas. A range of values is specified by two values separated by a colon. The values must be listed in ascending order.

New control variable values as shown in the second example, can be defined from overlapping values of the old variable. Not all values of the old variable need be used.

The LABEL command can be used to assign print labels to the new variable and its values.

It is a good idea to verify complex definitions with the LIST command.

LABEL
LABEL Command
Function:

Assign print labels to new control and observation variables, and to categories of control variables.

Syntax:

LABEL variable-name value 'label'

variable-name
The name of the new variable specified in a DEFINE, COMPUTE, or POST COMPUTE statement.

value
The category value when assigning labels to a value of a control variable.

label
The print label enclosed in single quotes. A label may be up to 20 characters long and may contain any printing character.

Examples:

```
LABEL INCOME 'INCOME GROUP'

LAB REGION 1 'SOUTHWEST'
```

SELECT
SELECT Command

Function:

Choose a subset of the data file for inclusion in the table. The SELECT command is a global filter statement which is in effect for the entire table. Further selection of categories of control variables can be done with the STUB and HEADING commands.

Syntax:

SELECT IF|UNLESS condition AND|OR condition ...

IF|UNLESS
IF specifies that data is to be selected when a selection condition is true. UNLESS means that data is chosen when the condition is not true.

condition
The selection condition is a logical expression composed of control variables, observation variables, arithmetic expressions, numeric constants and relation operators of the forms:

variable constant arith. exp.	relation operator	variable constant arith. exp.

There are two combinations that are not allowed: an observation variable and a control variable on opposite sides of the relation operator, and a control variable and an arithmetic expression on opposite sides.

DISPLAY
DISPLAY Command

Function:

Display row or column percentages or counts in table. The default is COUNTS.

Syntax:

DISPLAY display-spec

display-spec
The display requested can be one or more of the following:
STUB|ROW %
Row percentages, where percentages for each row sum to 100%
HEADING|COLUMN %
Column percentages, where percentages for each column sum to 100%
COUNTS
Counts will be displayed. COUNTS is the default.

The base count for percentage calculations is the highest level of aggregation or total requested. If totals are not requested, then the base is calculated as the sum over each observation variable.

Examples:

DISPLAY ROW %

DIS COUNTS

KEEP
KEEP or SAVE Command

Function:

Store terminal image of a table in a line file.

Syntax:

KEEP|SAVE TABLE IN file name

file name
The name of a permanent or temporary MTS file to contain the table. The table is always written at the end of the file i.e., file name (LAST +1). If the file does not exist, it will be created.

Examples:

```
KEEP TABLE IN TAB.1

SAVE -T
```

In the first example, the table is stored in a permanent file named TAB.1, and in the second, in a temporary file, -T. The table can be displayed on the terminal with the MTS LIST command or with the editor PRINT command. See the FORMAT command for saving tables to be used in reports.

FORMAT
FORMAT Command

Function:

Insert *FORMAT commands and store table in an MIS file.

Syntax:
FORMAT IN file name.

file name
The name of a permanent or temporary MTS line file. The table is always appended to the end of the file. If the file doesn't exist, it will be created.

Examples:
FORMAT IN REPORT.M
FORM YREND.M

COMMANDS
COMMANDS

COLLAPSE -- define new control variables; synonym for DEFINE

COLUMN -- define columns of a table; synonym for HEADING

COLLAPSE -- define new control variables; synonym for DEFINE

COMPUTE -- compute a new observation variable

DEFINE -- define new control variables; synonym for COLLAPSE

FILE -- chose data file; synonym for USE

FORMAT -- store table as a *FORMAT manuscript

GO -- specification of table is complete; display requested table

HEADING -- define columns of a table; synonym for COLUMN

HELP -- displays help message; synonym for ?

KEEP -- store table in a file; synonym for SAVE

LABEL -- assign print labels to new variables

LIST -- display current table specifications

NOPROMPT -- ITG will not prompt

POST COMPUTE -- compute new observation variable after data has been aggregated for table

PROMPT -- ITG will ask for table specifications

PRINT -- print a copy of table on line printer

REPLACE -- change previously entered command; synonym for RESET for REPLACE

RESET -- change previously entered command; synonym for REPLACE

ROWS -- define rows of table; synonym for STUBS

SAVE -- store table in a file; synonym for KEEP

SELECT -- chose a subset of data file

STOP -- terminate ITG

STUBS -- define rows or stubs of a table; synonym for ROWS

COMMAND SYNTAX
COMMAND SYNTAX

ITG commands consist of a command keyword, or a command keyword followed by one or more operands as needed. Operands are composed of operand key words and parameters, such as names and text, chosen by the user as appropriate.

A slash, "/", may precede the command keyword, and if ITG is operating in PROMPT mode, the slash must be used to override the prompt and allow entry of a different command.

A complete description of each command is available by entering "HELP command key word", e.g. "HELP FILES". The help descriptions use the following conventions.

1. All keywords are shown in uppercase. In order for ITG to recognize a command keyword at least the characters underlined must be entered. In prompt mode certain command key words are supplied by ITG, (they will appear on the terminal), and should not be entered.

2. Information that varies with each table and must be supplied by the user is shown in lowercase. Square brackets, " ", enclose items that are optional and are to be used as needed.
3. Curly brackets, " ", enclose a choice of two or more items. Choices are separated by vertical bars.
5. Ellipses, "...", mean that an operand can be repeated as needed.

Example:
SELECT IF |UNLESS condition1 AND|OR condition2 ...

"SELECT" is the command keyword. The characters, "SEL", are the minimum string that must be entered.

" IF |UNLESS " begins the operand portion of the command. Here a choice of two parameter keywords is indicated. The square brackets around "IF" indicated that "IF" is the default choice and "IF" need not be entered.

"condition 1" is a parameter to be supplied by the user. Such parameters are fully explained in the help descriptions.

" AND|OR condition 2 " shows a second optional segment of the operand. Either "AND" or "OR" must be entered.

"..." shows that additional segments of the operand, " AND|OR condition n ", can be specified.
Some real commands would look like this:

SELECT IF INCOME EQ 0

SEL UNLESS AGE GT 50 AND STATE EQ 0

END