AJD-102
CRT-902

TRIO Tutorial, Version 2.0

by

Marc Gaudry, Pierre Lestage
Jacques Guélat and Pierre Galvan
with
the collaboration of
Louis-Philippe Duclos and Francine Dufort

Agora Jules Dupuit, Département de sciences économiques, Université de Montréal, C.P. 6128, Succursale Centre-ville, Montréal, Canada H3C 3J7.

This paper is based on work financed in part by Transport Canada, and which benefitted directly or indirectly from research funded by the National Sciences and Engineering Research Council and by the Social Sciences and Humanities Research Council of Canada, as well as by the *Actions Structurantes* and FCAR programs of the Quebec Government. Isabelle Marullo and Caroline Barbaud made contributions as research assistants. The authors also thank Juan de Dios Ortuzar for making the Santiago database available.

Agora Jules Dupuit — Publication #102
Centre de recherche sur les transports — Publication #902

April 1993 — Revised in February 1994, September 1995 and September 2005

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 1993

Abstract

This manual is part of the TRIO regression program documentation listed on the last page of this text. It conducts the TRIO user through a first session with TRIO. Principal steps covered include: creation of a database, model estimation with LEVEL, SHARE and PROBABILITY algorithms, use of the TABLEX interactive table to examine regression results, and production of graphs.

Key-words: TRIO regression program, ordinary least squares regression, linear logit regression, Box-cox transformations, TABLEX table, graphic analysis.

Résumé

Ce texte fait partie de la documentation du progiciel d'analyse de régression TRIO dont on trouvera la liste en dernière page de ce manuel. Le *tutorial* donne une première idée du progiciel en faisant un survol de ses diverses capacités qui sont parcourues rapidement dans le cadre d'une première session TRIO. On exécute un certain nombre de tâches, dont la création d'une banque de données, l'estimation par des algorithmes de NIVEAUX, de PARTS et de PROBABILITÉS, l'usage du tableau interactif TABLEX destiné à l'analyse des résultats de régression et la production de graphiques.

Mots-clés : progiciel TRIO, régression par moindres carrés ordinaire, régression logit linéaire, transformations Box-Cox, tableau TABLEX, analyse graphique.

TRIO TUTORIAL DATE OF LAST MODIFICATION	
TITLE OF CHAPTER	DATE OF LAST REVISION
1: A FIRST TRIO SESSION	July 5, 2005
2: LEVEL CLASS	July 5, 2005
3: PROBABILITY CLASS	Feb 14, 1994
4: SHARE CLASS	July 26, 1994
5: INTRODUCTION TO TRIO GRAPHS	Nov 30, 1993
APPENDIX A: TRIO Word Editor Commands	Sept 27, 1995
APPENDIX B: TRIO Editor Commands	Nov 30, 1993
APPENDIX C: TRIO Graph Commands	Mar 10, 1995

OTHER TRIO DOCUMENTATION . . . . . See last page

Contents

CHAPTER 1: A FIRST TRIO SESSION

1.1 THE DEMONSTRATION DATA

Three sets of data are supplied with this Tutorial. We shall describe them briefly. They will be used to illustrate TRIO capabilities, in particular to estimate the parameters of variants, many of which correspond to published results that are also found in the Model Types manual.

1.1.1 The Montreal time series data (for LEVEL models)

The first set of data pertains to the area served by a Canadian transit company, the MUCTC (the Montreal Urban Community Transit Commission), and consists of variables that describe the total monthly transit ridership by adults and schoolchildren, as well as various explanatory factors or determinants of transit demand over the period December 1956 to December 1972 (193 observations). Explanatory variables belong to various categories: prices and vehicle availability, service levels, weather, economic activities, and so on. Because the MUCTC serves roughly 80% of the Montreal Metropolitan Area population, all variables are defined over that area.

1.1.2 The Santiago disaggregate data (for PROBABILITY models)

The second set of data pertains to a sample of 622 work trips by individuals taken in the Las Condes corridor of Santiago, Chile, in 1983, and was made available to us by the Department of Transportation Engineering of the Catholic University of Chile. The destination of all trips was the Central Business District. For each individual, we know the chosen transportation mode and which of the 9 modes were available. We also know the fare and service level characteristics of the modes, as well as a number of socio-economic characteristics of the individuals.

1.1.3 The Canada aggregate data (for SHARE models)

The third set of data pertains to a sample of 155 origin-destination pairs of Canadian cities in 1976. For each market or origin-destination pair, we know the market share of each transportation mode and which of the 4 transportation modes were available. We also know the fare and service level characteristics of the modes, as well as a number of socio-economic characteristics of the markets. The travel flows for the air and rail modes were obtained from differents carriers, the bus flows were obtained from the number of tickets sold and, for the car passanger trips, the flows were estimated by a trip distribution model calibrated using link flows. The network time data obtained from Transport Canada were derived from assignment algorithms and the frequencies computed from carrier schedules. Finally the socioeconomic data were obtained from the 1976 Canadian Census (Statitics Canada).

1.2 CREATE A DATABASE

In order to illustrate a first TRIO session, a special directory has been created during the TRIO installation, namely **C:\Program Files\Trio2X\demo**, which contains the data files associated with the three databases previously described:

1. The variables files: d111.in, d121.in and d131.in.
2. The groups files: g111.in, g121.in and g131.in.
3. The variants files: d211.in, d221.in and d231.in.
4. The labels files: l410.in, l420.in and l430.in.

To create a database using the files in "demo" involves the following steps:

1. Launch TRIO by double-clicking the "Run Trio" icon on the desktop.
2. Type a name, say "TestDemo", for the new database in the small textbox above the bigger window which lists all the databases already created. For a first TRIO session, this window is empty.
3. Click "Open" and answer "Yes" to the question "Do you want to copy some Trio data files to newly created Trio Data Bank"
4. By default, a new window displays all the data files in "demo". Select them all.
5. After importing all the data files into the new database which is stored in **C:\Program Files\Trio2X\Data\TestDemo**, answer "No" to the question "Do you want to copy some Trio data files to newly created Trio Data Bank"
6. Now, the module 0.01 INITIALIZE DATABASE is activated. This module which is called every time you create a new Trio database is used to set the size of your database.

```
0.01   INITIALIZE DATABASE

TYPE of variable referred to in the following questions
LEVEL

          Enter: Maximum number of periods ( 1) = 200
```

IMPORTANT NOTE: Throughout the Tutorial, you will notice that TRIO is an interactive software. All user's answers will be highlighted in bold style. So **the above answer would be 200**.

• *The* **Enter question** *always begins with the string "Enter:" followed by the description of the item(s) wanted. Either of two formats is used.*

The first format is:

Enter: <Kind of data expected> =

In this case, only one item is asked: the equal sign is on the same line. In many cases, a default value is provided for a single-item Enter question. It appears between parentheses and is selected by typing carriage return <CR>. Otherwise, a number or a character string must be entered. The input is always terminated by a carriage return <CR>.

The second format is:

Enter: <Kind of data expected><maximum items per line>
= <answer 1>
= <answer 2>
=

In this case, the equal sign stands on the next line. An answer with multiple entries is terminated by entering a blank or nothing. When more than one item may be entered on the same line, a maximum is displayed. The items must be delimited by a ***separator*** *which may be a blank, a comma or a semi-colon. Items exceeding the permitted maximum per line are neglected by the program.*

The default value shown on the previous page may be different from the one you see on your screen. Default dimensions depend on the version of TRIO you are using. Since the Montreal time series pertain to 193 periods, **enter 200** to leave a gap for future series updates. The value in parentheses is the default value.

```
SHARE

          Enter: Maximum number of observations ( 1) = 160
```

The second Enter group of questions uses the word "observations", which refers to share variables. The Canada share variables pertain to 155 observations. For that reason you should **enter 160**.

```
 Enter: Maximum number of alternatives on observations (   25) = 4
```

As the Canada database contains 4 alternatives (or modes), **enter 4**.

```
PROBABILITY

          Enter: Maximum number of individuals ( 1) = 700
```

The third Enter group of questions uses the word "individuals", which refers to probability variables. The Santiago probability variables pertain to 622 individuals so **enter 700**.

```
 Enter: Maximum number of alternatives on individuals (   25) = 9
```

As the Santiago database contains 9 alternatives, **enter 9**.

```
ALL TYPES

          Enter: Maximum number of variables per group (  100) =
```

Use the default value: **type <CR>**.

```
          Provide maximal sizes for dynamic data? n
```

• *The* **Yes/No question** *is used to select between two exclusive alternatives. It is always terminated by a question mark. No default value is provided. Possible answers are: "yes" or the letter "y", "no" or "n", "restart" or "r" to restart at the beginning of the module and finally "quit" or "q" to quit the module. Restart and quit are used to abort the current action. The input is always terminated by a carriage return <CR>.*

The user may, with this option, put limits on the disk space used by the database. The dynamic files grow when data are added to the database. In order not to exceed the allocated space, each type of data is limited. If you answer "yes" to the preceding question, you will be asked for the maximum of each data type. At the end, the maximal size of your database will be displayed. Always keep a gap between the total size and the disk space limit to allow space for scratch files used by TRIO. If you are not limited on disk space answer "**no**".

A summary of the previous dialogue is now displayed and you are asked to agree or not with the values previously entered. Look at those values carefully each time you create a new database because it is not possible to modify those values once the database is finally created. Once the database is created, if the user wants to modify those size values (say for example to have 12 alternatives instead of 9) the user will have to create a new database in another directory. For the purpose of this Tutorial the sizes entered are correct, so the user should **enter "yes"**.

```
 Enter: Project title (max. 80 chars)
------------------------------------------------------------------------

Tutorial Demonstration Database
```

Enter a brief description of the use of the database. Here, for example, one can enter "**Tutorial Demonstration Database**". When you've entered your title, Trio will ask for a last confirmation.

• *With the* **Select question,** *the user has to choose between many alternatives. No default value is provided. Each selection is composed of a number together with its associated item. The item is chosen*

by its number. The allowed answers are: the number of the selected item, "r" for restart or "q" for quit. The input is always terminated by a carriage return <CR>.

Now it is time to select the type of terminal your will be working with. If you are working with a Station type of computer (SUN Station), the following table will be displayed:

```
Read device table
     Device   1: Tektronix 4014
     Device   2: Tektronix 4115B
     Device   3: Emulator EM4105 color
     Device   4: Emulator EM4105 monochrome
     Device   5: Seiko GR-1104 (vt100)
     Device   6: Seiko GR-1104 (vt100) w. tablet
     Device   7: Visual 550 (vt100 mode)
     Device   8: SUN OPENWIN COLOR
     Device   9: SUN OPENWIN MONOCHROME
     Device  10: ANSI terminal 24x80 non-graphic
     Device  11: Dot matrix Printer (Letter)
     Device  12: Dot matrix Printer (A4)
     Device  13: Post-Script Printer (Letter)
     Device  14: Post-Script Printer (A4)
     Device  15: LaserJet Printer (Letter)
     Device  16: LaserJet Printer (A4)

Select: Type of terminal
        1 = Tektronix 4115B
        2 = Emulator EM4105 color
        3 = Emulator EM4105 monochrome
        4 = Seiko GR-1104 (vt100)
        5 = Seiko GR-1104 (vt100) w. tablet
        6 = Visual 550 (vt100 mode)
        7 = SUN OPENWIN COLOR
        8 = SUN OPENWIN MONOCHROME
        9 = ANSI terminal 24x80 non-graphic
```

However if you are working with a Personal Computer (386 with 387 coprocessor, or 486) then the choice list will be the following.

```
Read device table
     Device   1: IBM PC Color + Monitor VGA-25
     Device   2: IBM PC Mono + Monitor VGA-25
     Device   3: IBM PC Color + Monitor VGA-50
     Device   4: IBM PC Mono + Monitor VGA-50
     Device   5: Dot matrix Printer (Letter)
     Device   6: Dot matrix Printer (A4)
     Device   7: Post-Script Printer (Letter)
     Device   8: Post-Script Printer (A4)
     Device   9: LaserJet Printer (Letter)
     Device  10: LaserJet Printer (A4)
     Device  11: Tektronix 4014

Select: Type of terminal
        1 = IBM PC Color + Monitor VGA-25
        2 = IBM PC Mono + Monitor VGA-25
        3 = IBM PC Color + Monitor VGA-50
        4 = IBM PC Mono + Monitor VGA-50
```

Then you must select the type of printer you will be using.

```
Select: Type of printer
        1 = Dot matrix Printer (Letter)
        2 = Dot matrix Printer (A4)
        3 = Post-Script Printer (Letter)
        4 = Post-Script Printer (A4)
        5 = LaserJet Printer (Letter)
        6 = LaserJet Printer (A4)
```

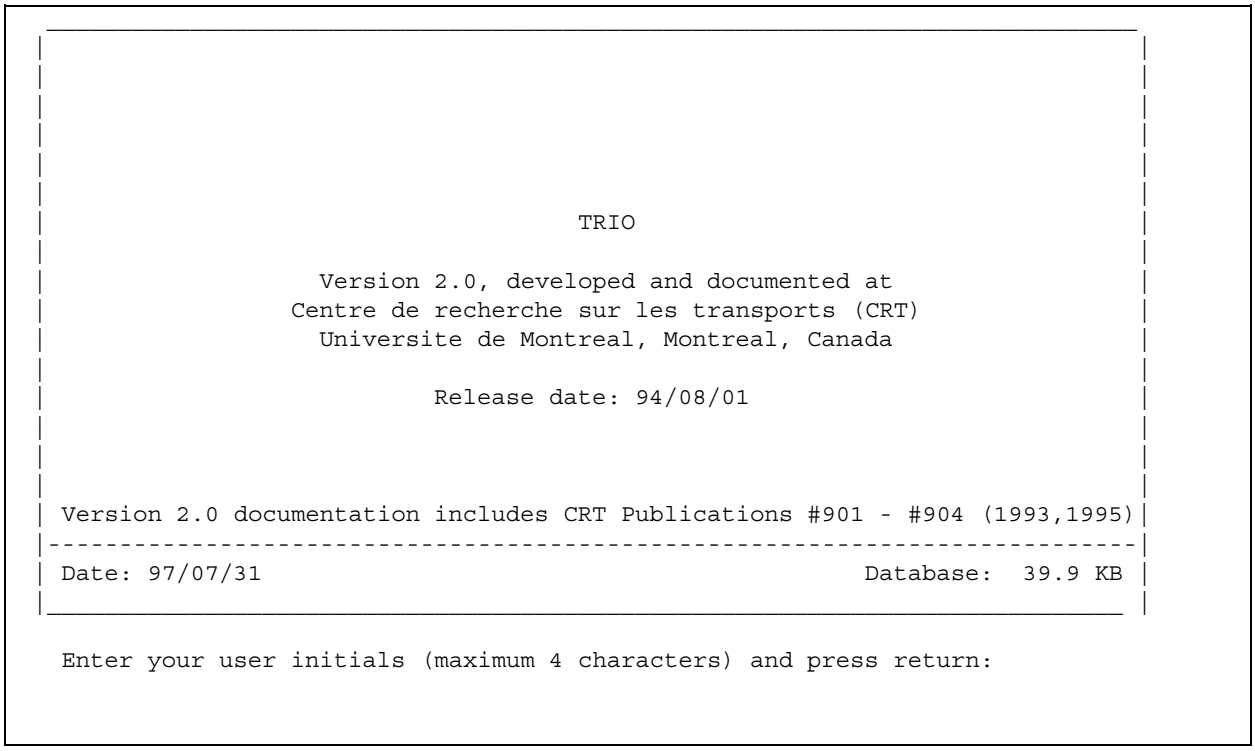
When graphs are to be sent in output, which may be on a printer, a plotter or a laser printer, the format of the input for those devices may differ. The next selection defines this device.

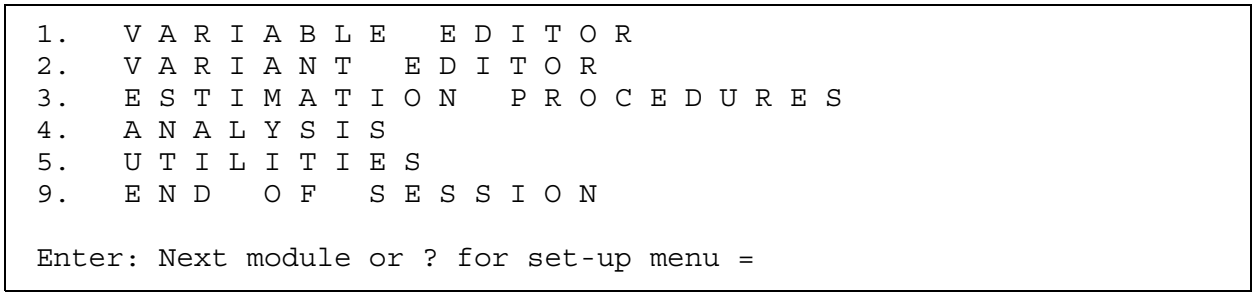
```
Select: Type of plotfile
        1 = Tektronix 4014
        2 = Tektronix 4115
```

You should note that these device selections are not permanent: you can modify these choices whenever necessary, during a TRIO session.

The database is finally initialized and ready to begin a TRIO session.

Every time someone uses this database, initials are asked in order to identify the person who brought modifications to the database.





Before ending this first Trio session using module “9”, you should read all the data files (.in) into TRIO to be able to work with these files. The next three chapters (Level, Probability and Share classes) will explain the reading procedures in detail.

Note that for later Trio sessions, to reopen a database already created, just launch TRIO and double-click a database of your choice in the big window below the small textbox.

CHAPTER 2: LEVEL CLASS

2.1 LEVEL VARIABLES

Information like the Montreal time series or the Santiago disaggregate cross section represents large amounts of data. The best way to manipulate the data is by using an ordinary ASCII file, thus avoiding the possible incompatibility between different computers. TRIO uses this approach to capture a large amount of data. An ASCII file is read in free format using a minimum of syntax to detect errors. Preparing the batch input file is done outside TRIO.

2.1.1 Reading the level variables input file

Select the VARIABLE EDITOR by entering "**1**". You should see each sub-module inside module 1.

```
1. V A R I A B L E   E D I T O R
     1.10 Edit level variables           1.11 Read/write level variables
     1.20 Edit share variables           1.21 Read/write share variables
     1.30 Edit probability variables     1.31 Read/write probability variables
2. V A R I A N T   E D I T O R
3. E S T I M A T I O N   P R O C E D U R E S
4. A N A L Y S I S
5. U T I L I T I E S
9. E N D   O F   S E S S I O N

Enter: Next module or ? for set-up menu =
```

The first demonstration file d111.in can now be read. **Select module 1.11** by typing 1.11. The module 1.11 menu looks like this:

```
1.11  READ/WRITE LEVEL VARIABLES

Select: 1 = Read variables from ascii file
        2 = Read groups from ascii file
        3 = Write variables to ascii file
        4 = Write groups to ascii file
        5 = End
         1
```

Select the **"Read variables from ascii file"** (option 1). TRIO then asks for the name of your input file in order to read the data. Press the "**Return**" key to select the default input file:

```
Enter: Name of your input file (d111.in) =
```

Then TRIO asks if you want to echo the batch reading.

```
Echo the batch reading? n
```

Answering yes allows you to see every line read from the input file. When the input file is quite large, as in our case, it is better not to echo the reading. The answer should then be "**no**".

Then a short report on the execution is displayed on your screen.

```
 Variable "ant" read.
 Variable "antl1" read.
 Variable "cardex" read.
 Variable "cheat2" read.
 Variable "csfmr" read.
 Variable "eanfp" read.
 Variable "ex0" read.
   .    .    .   .
 Variable "tlk" read.
 Variable "tt" read.
 Variable "tw" read.
 Variable "wd" read.
 Variable "xcanfa" read.
 Variable "xcscnfa" read.
 Variable "za" read.
 Variable "zsc" read.
WAIT. Press any key to continue.
```

When the read process terminates, you can exit **module 1.11** by typing the key "**q**" or by selecting option **5**.

We just saw how to enter variables into the database by reading an external batch input file. However, batch entry is an old-fashioned way of processing data and is only efficient when reading variables for the first time or when the number of variable elements to enter is very large. This is why TRIO provides the user with different interactive tools permitting an efficient and reliable management of the set of variables in the database.

2.1.2 List of variables

The first tool allows us to make a list of the variables included in the database. To do so **enter module 1.10**, select "**Edit variables**" and press **Return**. Then select "**List**" (option 4) and press again **Return**.

```
 Select: 1 = Edit variables
         2 = Edit groups
         3 = End
           1


 Select: Operation on variables
         1 = Add
         2 = Modify
         3 = Delete
         4 = List
         5 = Edit comments
           4

 Detailed list? n

 ant           antl1         cardex        cheat2        csfmr         eanfp
 ex0           ex1           ex7           ex8           ex9           fad
 fir           fscd          nc            p             rawem         rcutp
 rfk           rscmp         sat           sfk           shd           shonf
 str           struct1       t2t           tend          tlk           tt
 tw            wd            xcanfa        xcscnfa       za            zsc

WAIT. Press any key to continue.
```

A short list of variable names currently included in the database can be obtained by answering "**no**" to the "Detailed list?" question. The listed names are those of the level variables that may be used by TRIO.

Return again to module 1.10 main menu by pressing **<CR>**. Then select 1 "**Edit variables**" followed by option 4 "**List**" once more. More information on variables is obtained by requesting a detailed list. After answering "**y**" to "Detailed list?" you get the following question:

```
 List by group? n
```

• ***Groups*** *of variables are used to gather variables in a predefined order. A description is associated with each group. The groups are generated by the user and are used only as a utility. For example, when making a table or a list, the group name may be used to specify all the variables of that group.*

Since we have not created groups yet, answer "**no**".

```
 Do you want only the descriptions? n
```

This option allows to list the name and the two descriptions associated with each variable or the main statistics associated with it. For the moment, answer **"no"**.

```
 Enter: Name of variable(s) to be listed (max. 1000 characters)
 = *
```

In this context, the user is asked for one or more names (note that the equal sign is not on the line following the question). The string "(s)" following "variable" is indicating that one or more names may be entered. Since names of existing variables are requested, wild cards may be used.

• *The* **wild cards** *are used to abbreviate variables names using certain special characters. Those characters are: "*" and "?"; ? stands for any character and * for an arbitrary number of characters. For example "c*" stands for all variables beginning by c.*

Enter "*****" to list all the variables and press **<CR>**. A good exercise, if you are not familiar with wild card characters, is to generate lists of variables using different combinations. For example try "t*", "t?" and "*t*".

```
Select: List device
        1 = Terminal
        2 = Print file
          1
1 page(s) sent to terminal.
```

For the purpose of the exercise, type **"1"** and the list will be sent to the terminal. However, detailed reports may be sent to the reports file by choosing option "**2**". The reports file is used as default device for the print file.

```
TRIO module: 1.10                    Date: 93/01/26                    User: ....                    Page:    1
Project: Tutorial Demonstration Database
Data type: level
-------------------------------------------------------------------------------------------------------------------------------


List of level variables
Variable name Status Last mod. F. per. L. per.  Miss.  Min. value   Max. value   Mean value  Stand. dev.  Date of observ. (1)
_____________ ______ _________ _______ _______  _____  __________   __________   __________  ___________  ____________________

ant             V    93/01/05      1     193      0    0.0000E+00   0.1000E+01   0.5181E-02   0.7198E-01     0/ 1/ 1  0: 0: 0
antl1           V    93/01/05      1     193      0    0.0000E+00   0.1000E+01   0.5181E-02   0.7198E-01     0/ 1/ 1  0: 0: 0
cardex          V    93/01/05      1     193      0    0.0000E+00   0.1000E+01   0.1036E+00   0.3056E+00     0/ 1/ 1  0: 0: 0
cheat2          V    93/01/05      1     193      0    0.0000E+00   0.1000E+01   0.1399E+00   0.3478E+00     0/ 1/ 1  0: 0: 0
csfmr           V    93/01/05      1     193      0    0.0000E+00   0.1476E+03   0.2455E+02   0.3378E+02     0/ 1/ 1  0: 0: 0
eanfp           V    93/01/05      1     193      0    0.5098E+08   0.1465E+09   0.1234E+09   0.1775E+08     0/ 1/ 1  0: 0: 0
ex0             V    93/01/05      1     193      0    0.0000E+00   0.6851E+06   0.1066E+05   0.7746E+05     0/ 1/ 1  0: 0: 0
ex1             V    93/01/05      1     193      0    0.0000E+00   0.6727E+06   0.9531E+04   0.7277E+05     0/ 1/ 1  0: 0: 0
ex7             V    93/01/05      1     193      0    0.0000E+00   0.4250E+07   0.9473E+05   0.5553E+06     0/ 1/ 1  0: 0: 0
ex8             V    93/01/05      1     193      0    0.0000E+00   0.1061E+07   0.2135E+05   0.1288E+06     0/ 1/ 1  0: 0: 0
ex9             V    93/01/05      1     193      0    0.0000E+00   0.8849E+06   0.1232E+05   0.9192E+05     0/ 1/ 1  0: 0: 0
fad             V    93/01/05      1     193      0    0.1201E+00   0.2315E+00   0.1665E+00   0.3760E-01     0/ 1/ 1  0: 0: 0
fir             V    93/01/05      1     193      0    0.0000E+00   0.2200E+02   0.1140E+00   0.1584E+01     0/ 1/ 1  0: 0: 0
fscd            V    93/01/05      1     193      0    0.4805E-01   0.7323E-01   0.5946E-01   0.6512E-02     0/ 1/ 1  0: 0: 0
nc              V    93/01/05      1     193      0    0.1724E+06   0.5228E+06   0.3170E+06   0.9997E+05     0/ 1/ 1  0: 0: 0
p               V    93/01/05      1     193      0    0.1203E+03   0.1727E+03   0.1413E+03   0.1496E+02     0/ 1/ 1  0: 0: 0
rawem           V    93/01/05      1     193      0    0.6092E+02   0.1052E+03   0.8029E+02   0.1150E+02     0/ 1/ 1  0: 0: 0
rcutp           V    93/01/05      1     193      0    0.0000E+00   0.7847E+05   0.2410E+05   0.1988E+05     0/ 1/ 1  0: 0: 0
rfk             V    93/01/05      1     193      0    0.0000E+00   0.2058E+01   0.7273E+00   0.5056E+00     0/ 1/ 1  0: 0: 0
rscmp           V    93/01/05      1     193      0    0.0000E+00   0.4800E+06   0.3160E+06   0.1540E+06     0/ 1/ 1  0: 0: 0
sat             V    93/01/05      1     193      0    0.4000E+01   0.5000E+01   0.4352E+01   0.4789E+00     0/ 1/ 1  0: 0: 0
sfk             V    93/01/05      1     193      0    0.0000E+00   0.1797E+02   0.2664E+01   0.3835E+01     0/ 1/ 1  0: 0: 0
shd             V    93/01/05      1     193      0    0.4000E+01   0.7000E+01   0.5269E+01   0.8228E+00     0/ 1/ 1  0: 0: 0
shonf           V    93/01/05      1     193      0    0.1108E+06   0.4705E+06   0.2133E+06   0.6774E+05     0/ 1/ 1  0: 0: 0
str             V    93/01/05      1     193      0    0.0000E+00   0.2000E+02   0.2280E+00   0.1890E+01     0/ 1/ 1  0: 0: 0
struct1         V    93/01/05      1     193      0    0.0000E+00   0.1000E+01   0.3863E+00   0.4868E+00     0/ 1/ 1  0: 0: 0
t2t             V    93/01/05      1     193      0    0.9231E+01   0.1256E+02   0.1015E+02   0.8823E+00     0/ 1/ 1  0: 0: 0
tend            V    93/01/05      1     193      0    0.1000E+01   0.1550E+03   0.8932E+02   0.4654E+02     0/ 1/ 1  0: 0: 0
tlk             V    93/01/05      1     193      0    0.0000E+00   0.5300E+03   0.1942E+03   0.1451E+03     0/ 1/ 1  0: 0: 0
tt              V    93/01/05      1     193      0    0.1830E+02   0.2819E+02   0.2134E+02   0.1871E+01     0/ 1/ 1  0: 0: 0
tw              V    93/01/05      1     193      0    0.2276E+01   0.5159E+01   0.3972E+01   0.6567E+00     0/ 1/ 1  0: 0: 0
wd              V    93/01/05      1     193      0    0.1900E+02   0.2300E+02   0.2082E+02   0.1032E+01     0/ 1/ 1  0: 0: 0
xcanfa          V    93/01/05      1     193      0    0.9477E+07   0.3111E+08   0.2028E+08   0.2197E+07     0/ 1/ 1  0: 0: 0
xcscnfa         V    93/01/05      1     193      0    0.1195E+07   0.5624E+07   0.3071E+07   0.8138E+06     0/ 1/ 1  0: 0: 0
za              V    93/01/05      1     193      0    0.0000E+00   0.9797E-01   0.4532E-01   0.4067E-01     0/ 1/ 1  0: 0: 0
zsc             V    93/01/05      1     193      0    0.0000E+00   0.1991E+00   0.6562E-01   0.6218E-01     0/ 1/ 1  0: 0: 0


 Total:  36
```

Please note that for PC users the actual output displayed on the screen will differ in some cases.

Scrolling Commands:

During a scrolling session, the insert mode doesn't exist. The letters typed are used to perform movements of the cursor. Using "s" (left), "e" (up), "d" (right) and "x" (down) and you can move the cursor across the page (the commands "f", "t", "l" and "b", and allow for a quicker movement of the cursor across the page). TRIO also offers, to **PC users**, the possibility to use the direction keys to move around in the file instead of using the designated letters described above. Using "n" or "N" (next page), "p" or "P" (previous page), and "gx" (where "x" is the page number) allows the user to move through the pages of the document. As an exercise **try moving** the cursor around. Please **note** that for SUN users the only options during this scroling session will be "x", "e", "p" and "n" due to the full page presentation on the SUN station.

Finally you can quit the scrolling session with the command "q".

An echo of these commands is available during a scrolling session by using the command "h". For a **complete** definition of these commands refer to the Reference Manual.

After the scrolling session TRIO will always give you the choice to print your list or table in an output file of your choice. If you are using a **SUN** computer, the file name can't exceed 12 characters. If you are using a **PC**, the file name must follow the **DOS** rules (eight characters followed by a dot and a three character file extension).

```
Send to print file? no
```

If you answer "**no**", the preceeding table won't be saved in a report file and you will return to the module main menu. If you type "**yes**", the following message will appear:

```
Enter: Name of your output file (reports) =
```

The reports file is set by default but you could rename the file by choosing any name. For example, you could use the name "**variables**" and this file will be saved in the reports directory associated with the database and will be named "**variables**".

Now we are going to repeat the exercise by sending the output to the print file. The dialogue of the entire operation follows:

```
1.10  EDIT LEVEL VARIABLES

Select: 1 = Edit variables
        2 = Edit groups
        3 = End
          1


Select: Operation on variables
        1 = Add
        2 = Modify
        3 = Delete
        4 = List
        5 = Edit comments
          4

Detailed list? y

List by group? n

Do you want only the descriptions? n

Enter: Name of variable(s) to be listed (max. 1000 characters)
= *

Select: List device
        1 = Terminal
        2 = Print file
          2

Enter: Name of your output file (reports) = <CR>

1 page(s) sent to print file.
```

Please note that since one may accumulate many reports in the same file, the page number is not reset to zero at every report. To modify the page counter you must enter main **module 000**. Exit the actual TRIO session by selecting "**End**" (3) then quit TRIO by selecting **module 9**, begin a new one by typing "**TRIO**" and then answer "**?**" or "**<CR>**" to the question "Enter: Next module or ? for set-up menu =" .

```
Enter: Next module or ? for set-up menu = ?

Select: Next command
        1 = Call a module
        2 = List available module groups
        3 = List available modules
        4 = Redefine current page number
        5 = Get devices types
        6 = Set,reset and display the switches
        7 = Display database parameters
        8 = Change project title
          4
```

Select the "*Redefine current page number*" option and enter the page number you want TRIO to use when numbering the next report pages. For example enter "**1**". The other options are explained in the Reference Manual.

```
Enter: Page = 1

Current page no.:   1

Enter: Next module or ? for set-up menu =
```

After quitting a module of TRIO (1.10 for example), the command "?" gives you the possibility to change only the device types. If you want to enter the command menu, you must press <CR> and return to the main menu of TRIO and then you must press <CR> one more time. If you press "?" to the first "Enter" question after having left a module the following screen will appear:

```
Enter: Next module or ? to change device = ?

Current terminal type is: your type of terminal
Change? n

Current printer  type is: your type of printer
Change? n

Current plotfile type is: your type of plotfile
Change? n
```

2.1.3 Creating new variables

There are many ways of creating new variables: you may define the new variable

1. by entering the values for different observations;
2. as an algebraic expression made up of existing variables;
3. by modifying an existing variable, first copying the variable and then modifying the copied values;
4. as a by-product of a variant estimation.

In the case of level variables, supplementary operations such as differentiation are also available. In this section, an example of creation will be presented using the notion of algebraic expression.

2.1.3.1 Algebraic expressions

TRIO provides the user with a powerful variable creation tool: a built-in algebraic expressions calculator. A new variable may be defined as an algebraic expression of existing variables. Let us immediately proceed with a very simple example: imagine that you wish to define a **constant variable**. In module 1.10, you select successively the option 1 ("Edit variables") and 1 ("Add"), then choose from the "Creation method" menu option number 4; "**Define algebraic expression**".

```
Select: Creation method
        1 = Copy a variable
        2 = Enter interactively
        3 = Take the difference of a variable
        4 = Define algebraic expression
        5 = Take from variant results
          4

Enter: Name of variable to be created = one

Enter: Algebraic expression
= 1
=

Select: 1 = Edit  a comment
        2 = Copy  a comment
        3 = Print a comment
        4 = List  a comment
        5 = Change table type
        6 = List   table contents
        7 = No further edition
          7

Variable "one" now exists in database.
```

If you list this variable, you will see that it is defined on the entire allowed period [1,200]. This will always be the case when no variable appears in the algebraic expression. If, however, you want to define a constant variable on a restricted period interval, you will have to use an existing variable in the algebraic expression. The next example shows how to create a dummy variable associated with variable "za". Select "**Define algebraic expression**" once again.

```
Enter: Name of variable to be created = za_dummy

Enter: Algebraic expression
= za.eq.0
=
```

The definition interval of "za_dummy" is the same as that of variable "za". Generally, the interval of definition of an algebraic expression is the intersection of the definition intervals of the variables appearing in the expression. This example illustrates the use of a logical **operator.** The value of a logical expression is one when true and zero when false. For the list of the available operators please refer to the Reference manual.

Let us continue our demonstration by defining another algebraic expression.

```
Enter: Name of variable to be created = complex

Enter: Algebraic expression
= (xcanfa.ge.20e6) * (sqrt(abs(rawem)) / (ln(tlk + one) + one)
=                       + sat ^ sin(tend.min.tlk))
=
```

As illustrated in the last example, algebraic expressions may be written on several lines. A maximum of **10** lines each containing **80** characters is permitted. As an exercise, please create the last two algebraic expressions on your own.

```
Enter: Name of variable to be created = mixed

Enter: Algebraic expression
= 1 +
= one +
= t2t +
= za_dummy +
= complex
=
```

```
Enter: Name of variable to be created = crowded

Enter: Algebraic expression
= rawem + sqrt(abs(complex)) + cos(mixed) + (tlk+tend)/2. + one
=
```

The defined algebraic expressions are kept in the database in a special file of formulas. You can always list the defined formulas by entering **module 5.10** and then choosing option **2** of the primary selection menu.

```
5.10  ALGEBRAIC EXPRESSIONS UTILITIES

Select: 1 = List of operators and keywords
        2 = List of algebraic expressions
        3 = List the tree of ancestor variables
        4 = List the tree of descendant variables
        5 = End
          2

Select: Type of variables
        1 = Level
        2 = Share
        3 = Probability
          1

Enter: LHS variable names (max. 1000 characters)
= *

Select: List device
        1 = Terminal
        2 = Print file
         1
```

```
TRIO module: 5.10      Page:   1                    Date: 93/11/05                                    User: ....
Project: Tutorial Demonstration Database
Data type: level
-------------------------------------------------------------------------------------------------------------------------------


   L I S T   O F   A L G E B R A I C   E X P R E S S I O N S
   _______   ___   _______________   ___________________

 one          = 1

 za_dummy     = za .eq. 0

 complex      = (xcanfa .ge. .2000000e+08) * (sqrt(abs(rawem)) / (ln(tlk +

                one) + one) + sat ^ sin(tend .min. tlk))

 mixed        = 1 + one + t2t + za_dummy + complex

 crowded      = rawem + sqrt(abs(complex)) + cos(mixed) + (tlk + tend) / 2

                + one
```

Now, if you want to visualize the links between the variables created during the demonstration you must select "**List the tree of ancestor variables**" in **module 5.10**.

```
5.10  ALGEBRAIC EXPRESSIONS UTILITIES

Select: 1 = List of operators and keywords
        2 = List of algebraic expressions
        3 = List the tree of ancestor variables
        4 = List the tree of descendant variables
        5 = End
          3

Select: Type of variables
        1 = Level
        2 = Share
        3 = Probability
          1

 Tree of variable ancestors

Enter: Variable name = complex
```

The tree list may be visualized in two modes: extended (ancestors of all variables are displayed) and condensed (ancestors of variables, that have not already appeared in an other level of the tree, are displayed). Select "**Extended**" (option 2).

```
Select: Type of tree list
        1 = Condensed
        2 = Extended
          2

Select: List device
        1 = Terminal
        2 = Print file
          1
```

The following table should appear on your screen:

```
TRIO module: 5.10      Page:   1                    Date: 93/11/05                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------


           E X T E N D E D   T R E E   O F   V A R I A B L E   A N C E S T O R S
           _____________   ______   __   _____________   _______________


-complex-----!-xcanfa
             !
             !-rawem
             !
             !-tlk
             !
             !-one
             !
             !-sat
             !
             !-tend
```

As you can see, using option 2 displays all the variables included in the right hand side of the algebraic expression. Exit now the list of algebraic expressions by entering the letter "**q**", answering "**no**" to the question "Send to print file?" and finally by choosing option "**5**" (End).

2.1.3.2 Taking the difference of a variable

As an example, we will take the difference of variable "csfmr", the monthly accumulated snowfalls at the Dorval airport of Montreal. The differential operator will be one year, that is twelve periods. To see how this type of operator works enter **module 1.10**, select "**Edit variables**" and then the option "**Add**". Then follow the instructions shown in the next table.

```
Select: Creation method
        1 = Copy a variable
        2 = Enter interactively
        3 = Take the difference of a variable
        4 = Define algebraic expression
        5 = Take from variant results
          3

Enter: Name of variable to be created = csfmr_diff12

Enter: Name of variable to be differentiated = csfmr

Enter: Differential operator = 12
```

The best way of getting a first idea of what variable "csfmr_diff12" looks like is to make a graph with it. Following the econometrics of time series, the variable should represent the deviations between the two periods. First enter **module 4.10** and select "**Make a graph**" option followed by **<CR>**.

```
Select: Graph type
        1 = Horizontal line graph
        2 = Vertical   line graph
        3 = Scatter         graph
        4 = Bar             graph
        5 = Pie             graph
          1

Enter: Graph title  = TAKING THE DIFFRENCE OF A VARIABLE

Enter: Character size [1 - 6: default = 2] =

Enter: Data sets to consider (1) = 2

Scale Y values? n
```

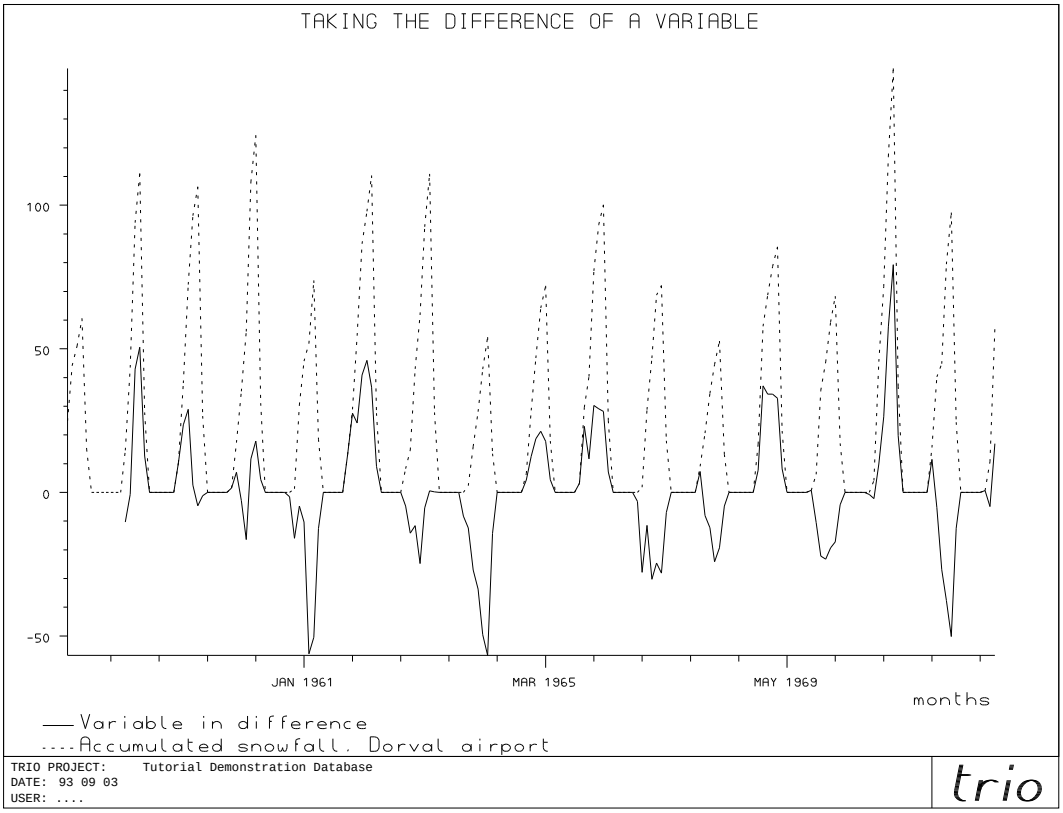
```
Enter: X axis description:(time) = months

Enter: Character size [1 - 6: default = 2] =

Do you wish to display labels for X values? y

Define Y axis values:

Select: Type of values
        1 = level variables
        2 = level variant results
          1

Enter: Algebraic expression = csfmr_diff12

Enter: Y axis description (csfmr_diff12) = Variable in difference

Enter: Character size [1 - 6: default = 2] =

Y values are defined for period   13 to   193

Enter: First period to consider =

Enter: Last period to consider =

Another Y variable? y

Define Y axis values:

Select: Type of values
        1 = level variables
        2 = level variant results
          1

Enter: Algebraic expression = csfmr
```

```
 Enter: Y axis description (ACCUMULATED SNOWFALL, DORVAL AIRPORT) =
Accumulated snowfall, Dorval airport

 Enter: Character size [1 - 6: default = 2] =

 Y values are defined for period    1 to   193

 Enter: First period to consider =

 Enter: Last period to consider =

 Save in the temporary list of graphs? y

 Plot? y
```

Press the key "**q**" to quit the graph. For additional information concerning TRIO graphic possibilities the user should refer to the Reference Manual where more examples are produced.

2.1.4 Modifying existing variables

The user may also wish to modify an existing variable without creating a new one. For example, this happens when a correction to an erroneous observation value has been detected or when variables are updated (new available observations). Modifying a variable does not imply losses of results in the database, except of course for modifications of the values of the variable (in this case, the consistency of the results of a variant using the modified variable could be altered and all the results of the variants using this variable in their structural definition are lost).



As an exercise the only modification that we will do is to modify the description page of a variable.

2.1.4.1 Modifying the description pages of a variable

The variable and variant descriptions are edited in TRIO with a special built-in full screen word editor and some of its capabilities are described here. For a complete list of the TRIO word editor commands you should refer to Appendix A.

As an example, we will show how to edit the description of variable "tt". Enter **module 1.10**, select "**Edit variables**" and then select the "**Edit comments**" option.

```
 Select: 1 = Edit  a comment
         2 = Copy  a comment
         3 = Print a comment
         4 = List  a comment
         5 = Change table type
         6 = List   table contents
         7 = No further edition
           1

 Enter: Variable name = tt
```

You will now enter the full page editor.

```
line:   3   col:  11   insert  ON    left:   7   page: 1   EMPTY

    type: level_______              variable name: tt__________
    user: ...._____________         date: 93/11/05

description1: MUCTC IN-VEHICLE TRAVEL TIME_____________________________________

description2: TEMPS DE TRANSPORT EN VEHICULE POUR LA STCUM_____________________
comments:
_________!_________!_________!_________!_________!_________!_________!_________!


A. OPERATIONAL DEFINITION OF TT
                  ______________________________

             A problem arises with respect to the speed of travel on public vehi
cles. If that speed were the same at any point of the city, or at any time, a si
ngle number for velocity V would, given the length of some representative trip L
*, yield an unambiguous single number TT=L*/V to denote average transit time. We
 have to assume that the spatial and temporal distribution of speed is optimal a
nd consider some measure of average speed as relevant.

             We shall use the number: TT = L*/V
```

At the beginning of an editing session, you are in positioning mode: the letters typed are used to perform movements of the cursor. Again using the keys "s","d","x","e" you will be able to move the cursor across fields. As an example pressing "**x**" six times moves the cursor into the comment field.

The command "i" is used to enter insertion mode. **Note that to exit any mode, you can always press the Escape key**. In a fixed field, such the name field, pressing "Return" exits from the insertion mode and positions the cursor at the next field. When you are in a field, the status line informs you of the number of characters left blank in that field. You cannot insert new characters in a filled field. Note that the page you are editing is a copy of the original. Modifications are effective only when the page is saved (using command "k").

As a first exercise, **delete the first line of the text**: "A. OPERATIONAL DEFINITION OF TT". Position the cursor on the line and press "**yl**". The line is deleted and the page is reformatted. Now insert

the new text: “A. GOOD DEFINITION OF TT”, first by entering the insert command “**i**” followed by the text and then by pressing the **Escape key “ESC”** in order to stop the insertion mode. Delete some of the underscore using the operator “**yy**”. Your comment should look like:

```
comments:
_________!_________!_________!_________!_________!_________!_________!_________!


A. GOOD DEFINITION OF TT
   ____________________
```

Always remember that a Help menu is available to the user by pressing the key “h”.

As a second example, we will modify the contents of the field “user”. Move the cursor to that field using the key “**e**” and set the insert mode off using “**v**”. The text you will enter now will overwrite the old one. Type your full first name, not exceeding 15 characters.

```
    user: I.M._____________            date: 93/11/05
```

To quit editing, you must first be in positioning mode. If you are entering text, first **exit** that mode by pressing “**Esc**”. To quit editing that page you type “**q**”, if you don’t want to save the page and by “**kq**” if you want to save the page. Let us quit without saving: type “**q**”.

```
QUIT EDITED PAGE? Y/N y
```

Answer yes (“**y**”) and you are back at the editor primary select. You may now practice on the text editor. A good habit is to keep a copy of Table A at hand. Remember, you may always quit a page without saving it by using command “**q**”.

2.1.5 Deleting variables

As for modifications of variable values, **deleting a variable may destroy consistency of variant results**. Worse, the variant definition itself may become invalid. For this reason, the deletion of a variable may imply the complete deletion of variants. In fact, all the variants where the deleted variable appears will be lost. **You should then be very careful before deciding to eliminate a variable from the database.**

Let us proceed with an example. We will delete variable “csfmr_diff12” that will not be used any more in this tutorial. Select option “**3**” of the “Operation on variables” menu.

```
Select: Operation on variables
        1 = Add
        2 = Modify
        3 = Delete
        4 = List
        5 = Edit comments
          3

Enter: Name of variable(s) to be deleted (max. 1000 characters)
= csfmr_diff12
=

Do you really want to delete these variables? y
Variable "csfmr_diff12" has been deleted.
```

Since variable "csfmr_diff12" did not appear in any variant, only one warning message has been issued before deletion was accomplished. TRIO wants to know if you agree to delete the variable.

2.2 DEFINE AND INITIALIZE LEVEL VARIANTS

We will now define our first experiment of the level class using the L-1.3 model type. To define our first experiment or, as we call it in TRIO, our first variant, enter **module 2.10**.

```
2.10  EDIT LEVEL VARIANTS

Select: 1 = Create     variants
        2 = Modify     variants
        3 = Initialize variants
        4 = Delete     variants
        5 = List       variants
        6 = See variant contents
        7 = Edit comments
        8 = End
          1
```

2.2.1 Creating a variant

Select the first option "**Create variants**". We have now the choice to create a reference variant by copying an existing one or by entering it interactively.

```
Select: Creation method
        1 = Copy a variant
        2 = Enter interactively
          2

Enter: Name of variant to be created = sch_bc1auhg
```

Our first experiment will explain the school-children demand with a non-linear regression model (that will also include an autocorrelation structure and heteroskedasticity variables). The name given

to the variant will be “sch_bc1auhg” (“bc1” for one Box-Cox transformation, “au” for Autocorrelation structure and “hg” for Heteroskedasticity variables).

```
                                VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
             NO BC

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|


HETEROSKEDASTICITY VARIABLES




AUTOCORRELATION STRUCTURE
```

You are now in the level variant editor. This is a full screen editor which uses a subset of the text editor commands. Again you may use the command key “**h**” if you want to obtain a Help menu describing the different commands available.

You are on the first field of the page: the dependent variable field (the user should consult the Model Types manual to find the exact meaning of each field of the variant editor). Using the command “**i**”, enter the name of the variable followed by <CR> or Esc. If the variable does not exist, you will get a warning message at the bottom of the page. For example one could have entered:

```
                            VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
mistake      NO BC

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|


HETEROSKEDASTICITY VARIABLES




AUTOCORRELATION STRUCTURE




 This variable does not exist!
```

In that case, after having read the message (This variable doesn't exist!), type "**<CR>**" to return in the field you were editing. You are still in insertion mode, so you do not have to type "**i**" again. Enter the right dependent variable name: "**xcscnfa**".

```
                          VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
xcscnfa      NO BC

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|


HETEROSKEDASTICITY VARIABLES




AUTOCORRELATION STRUCTURE
```

After the acceptance of a name, you are back in positioning mode. Again, using the same commands ,"**s**", "**e**", "**d**", "**x**" you can move from field to field. Go down to fill the fixed part variables. Type **"i"** to get in insertion mode and enter the variable names shown in next table. Every time a new variable name is entered, its default Box-Cox structure (no Box-Cox parameter) appears on the right. Use command "**d**" twice to move to the next field and enter the next variable. For the moment do not enter anything on the Box-Cox field. At the end of the edition you should have the following structure:

```
                          VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
xcscnfa      NO BC

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
fscd         NO BC  nc           NO BC  p            NO BC  rawem        NO BC
tw           NO BC  tt           NO BC  t2t          NO BC  tlk          NO BC
rfk          NO BC  sfk          NO BC  csfmr        NO BC  tend         NO BC
rscmp        NO BC  shonf        NO BC  ex7          NO BC  ex8          NO BC
zsc          NO BC  str          NO BC  wd           NO BC  sat          NO BC
shd          NO BC  ant          NO BC  antl1        NO BC  struct1      NO BC
cheat2       NO BC  cardex       NO BC

HETEROSKEDASTICITY VARIABLES




AUTOCORRELATION STRUCTURE
```

The words "NO BC" associated with each variable indicates that the variables are not Box-Cox-transformed. Nonlinear terms may be introduced in the models by using **Box-Cox Transformations** on the intervening variables. In the following example, the same unique Box-Cox parameter will be

assigned to the dependent variable and to a subset of the fixed part variables. For additional information concerning the Box-Cox transformations the user should refer to the Reference manual.

Imagine now that you want variable "nc" to be Box-Cox transformed in the model. Move to the corresponding field with the usual editor commands (**the Box-Cox parameter field of a variable is always at the right of the variable name**). Use now command "**ib**" to put a BC transformation on variable "nc".

```
                          VARIANT DEFINITION


DEPENDENT VARIABLE
|-----------------|
xcscnfa      NO BC

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
fscd         NO BC  nc           BC     p            NO BC  rawem        NO BC
tw           NO BC  tt           NO BC  t2t          NO BC  tlk          NO BC
rfk          NO BC  sfk          NO BC  csfmr        NO BC  tend         NO BC
rscmp        NO BC  shonf        NO BC  ex7          NO BC  ex8          NO BC
zsc          NO BC  str          NO BC  wd           NO BC  sat          NO BC
shd          NO BC  ant          NO BC  antl1        NO BC  struct1      NO BC
cheat2       NO BC  cardex       NO BC

HETEROSKEDASTICITY VARIABLES



AUTOCORRELATION STRUCTURE
```

In our example, we will only define one Box-Cox parameter for different variables. To do so, enter command "**i1**" followed by the "**<CR> or <ESC> key** on the corresponding Box-Cox fields shown below so that all the variables of group 1 will be transformed identically.

```
                          VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
xcscnfa      G  1

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
fscd          G   1  nc           G   1  p            G   1  rawem        G   1
tw            G   1  tt           G   1  t2t          G   1  tlk          NO BC
rfk           NO BC  sfk          NO BC  csfmr        NO BC  tend         G   1
rscmp         NO BC  shonf        G   1  ex7          NO BC  ex8          NO BC
zsc           NO BC  str          NO BC  wd           G   1  sat          G   1
shd           G   1  ant          NO BC  antl1        NO BC  struct1      NO BC
cheat2        NO BC  cardex       NO BC

HETEROSKEDASTICITY VARIABLES



AUTOCORRELATION STRUCTURE
```

As a second modification of variant "sch_bc1auhg", we will introduce **four Autocorrelation** parameters: lags of **1, 3, 4 and 12** months will be used.

In the autocorrelation structure part of the variant definition page, you have to introduce the autocorrelation orders to be considered. An order is entered by typing "**i**", followed by the order numbers "**1 3 4 12**" followed by a "**<CR> or the <ESC>**" key.

Finally we will introduce variable "**tend**" in the "**Heteroskedasticity variables**" section. Again, in the variant definition edition page, **place the cursor in the heteroskedasticity part** and introduce the variable "tend" ("**i**" followed by "**tend**" and **<CR>**):

At the end of the edition you should have the following structure:

```
                          VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
xcscnfa      G   1

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
fscd         G   1  nc           G   1  p            G   1  rawem        G   1
tw           G   1  tt           G   1  t2t          G   1  tlk          NO BC
rfk          NO BC  sfk          NO BC  csfmr        NO BC  tend         G   1
rscmp        NO BC  shonf        G   1  ex7          NO BC  ex8          NO BC
zsc          NO BC  str          NO BC  wd           G   1  sat          G   1
shd          G   1  ant          NO BC  antl1        NO BC  struct1      NO BC
cheat2       NO BC  cardex       NO BC


HETEROSKEDASTICITY VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
tend         BC


AUTOCORRELATION STRUCTURE
   1         3         4        12
```

Save the new variant using the command "**k**" and the message "This variant has been saved" will appear. The next question asked by TRIO will be:

```
 Initialize variant? n
```

You are now asked whether you wish to initialize it. For the moment answer "**no**". As for variables, every time a new variant is created you have the opportunity of editing its associated description pages (all versions of a same variant share the same description pages). Because this type of editor is very similar to the variables comment editor select "**No further edition**".

```
 Select: 1 = Edit  a comment
         2 = Copy  a comment
         3 = Print a comment
         4 = List  a comment
         5 = Change table type
         6 = List   table contents
         7 = No further edition
           7

 Variant "sch_bc1auhg:0" now exists in database.
```

Select the "**No further edition**" option and the system informs you that a new variant has been saved into the database. Version number **0** will be given by the system to this newly defined variant.

To see the available variants in the database, select the "**List variants**" option of the primary selection menu and then ask for a detailed list.

```
2.10  EDIT LEVEL VARIANTS

Select: 1 = Create     variants
        2 = Modify     variants
        3 = Initialize variants
        4 = Delete     variants
        5 = List       variants
        6 = See variant contents
        7 = Edit comments
        8 = End
          5

Detailed list? y

Enter: Name of variant(s) to be listed = *

Select: List device
        1 = Terminal
        2 = Print file
          1

   1 page(s) sent to the terminal.
```

Sending the reports to your terminal and using the scrolling, you should see the following:

```
TRIO module: 2.10      Page:   1                    Date: 93/11/05                                  User: ....
Project: Tutorial Demonstration Database
Data type: level
-------------------------------------------------------------------------------------------------------------------------------


List of level variants:

   Variant name   Version   Status   Date of last modification
   ____________   _______   ______   _________________________

   sch_bc1auhg        0        C           93/11/05


   Created:  1    Initialized:  0    Estimated:  0   Total:  1
```

In the list displayed on your screen, you can find the names and version numbers of the available variants.

2.2.2 Initializing a variant

The next logical step is to initialize the variant just created (initial values must be given in order to estimate the variant). Still in **module 2.10** select the "**Initialize variant**" option.

```
Enter: Name[:version] of variant to be initialized = sch_bc1auhg
```

Enter the name of our first variant "**sch_bc1auhg** ". Note that the version number is not specified because the default version number always corresponds to the one last created.

First you must define the estimation parameters. Refer to the Reference manual for more information about those parameters. For the purposes of the demonstration, **enter the proposed numbers (not the default numbers)**.

```
 Enter: First estimation period (   13) = 13

 Enter: Last estimation period (  193) = 181

 Enter: Last prediction period (  193) = 193

 Enter: Number of periods at end of sample
        used to evaluate elasticities (    1) = 12
```

After entering all the proposed values you will enter the variant "Initialization page". As an exercise move around the variant initialization page using the keys "**x**" and "**e**".

```
                    LEVEL VARIANT INITIALIZATION
   names          value          f/e          value         f/e
tlk           Conditional L-S    _        no box cox          _
rfk           Conditional L-S    _        no box cox          _
sfk           Conditional L-S    _        no box cox          _
csfmr         Conditional L-S    _        no box cox          _
tend          Conditional L-S    _         0.100000000E+01   E
rscmp         Conditional L-S    _        no box cox          _
shonf         Conditional L-S    _         0.100000000E+01   E
ex7           Conditional L-S    _        no box cox          _
ex8           Conditional L-S    _        no box cox          _
zsc           Conditional L-S    _        no box cox          _
str           Conditional L-S    _        no box cox          _
wd            Conditional L-S    _         0.100000000E+01   E
sat           Conditional L-S    _         0.100000000E+01   E
shd           Conditional L-S    _         0.100000000E+01   E
ant           Conditional L-S    _        no box cox          _
antl1         Conditional L-S    _        no box cox          _
struct1       Conditional L-S    _        no box cox          _
cheat2        Conditional L-S    _        no box cox          _
cardex        Conditional L-S    _        no box cox          _

HETEROSKEDASTICITY VARIABLE
tend           0.000000000E+00   E         0.100000000E+01   E

AUTOCORRELATION PARAMETERS
   1           0.000000000E+00   E
   3           0.000000000E+00   E
   4           0.000000000E+00   E
  12           0.000000000E+00   E
```

Since in our example we want to keep the default settings, end this edition page with command "**k**".

NOTE: the constant term doesn't appear in the table because its associated β cannot be modified and no Box-Cox parameter can be associated with this variable. However, this variable will be considered in the estimation phase.

It is important to use "**k**" to quit this edition page because if you use "q" TRIO will end the initialization procedure and return you to the primary selection menu without saving the modifications.

The last group of parameters you must define is the maximization algorithm parameter group. **Select the default parameters of the maximization procedure by answering <CR> to the following questions**:

```
Enter: Maximum number of iterations in likelihood
       function maximization algorithm ( 100) =

Enter: Tolerance factor in likelihood function
       maximization algorithm (.10E-04) =

Enter: Upper limit of Y for the computation of E(Y)
       and of the mean prob that Y exceeds this limit
       if lambda(Y)<0  (.00E+00) =

Enter: Initial value for diagonal of the
       approximate Hessian ( 1.0000) =

Save initialization? y

Initializing variant ...
```

When all the parameters of the model and of the estimation procedure have been defined, the initialization phase starts. Initial results such as the loglikelihood function initial value are computed. Depending on the size of the variant (number of variables and parameters), this operation may take some time to complete; so wait until the next prompt appears, without pressing any key in the meantime.

Once the variant is initialized you may produce a list of the variants included in the database.

```
List of level variants:

   Variant name   Version   Status   Date of last modification
   ____________   _______   ______   _________________________

   sch_bc1auhg        0        I            93/11/05

   Created:  0    Initialized:  1    Estimated:  0   Total:  1
```

Quit the table using "**q**" and by answering "**no**" to the printing question.

2.2.3 Starting a new series of experiments

The Montreal data has been collected to analyze total monthly transit ridership by schoolchildren and adults. Hitherto we have only designed an experiment describing schoolchildren ridership. The next experiment will concern adult ridership. We will start from the schoolchildren experiment and modify it to the adult context. The faster way of doing this is to copy the last "school" (sch_bc1auhg:0) variant and give it a new name (ad_bc2auhg:0) by using the "**Copy a variant**" command. To do so please follow the instructions shown in the next two tables.

```
2.10  EDIT LEVEL VARIANTS

Select: 1 = Create     variants
        2 = Modify     variants
        3 = Initialize variants
        4 = Delete     variants
        5 = List       variants
        6 = See variant contents
        7 = Edit comments
        8 = End
          1

Select: Creation method
        1 = Copy a variant
        2 = Enter interactively
          1

Enter: Name of variant to be created = ad_bc2auhg

Enter: Name[:version] of variant to be copied = sch_bc1auhg

Variant "ad_bc2auhg:0" now exists in database.
```

In effect variants created by the copying mechanism only differ from the original ones by their name (they keep the same status of the original variants; C, I or E). Also note that only modified variants, or variants interactively, can be initialized.

Now you must modify variant “ad_bc2auhg“ in the following way:

```
2.10  EDIT LEVEL VARIANTS

Select: 1 = Create     variants
        2 = Modify     variants
        3 = Initialize variants
        4 = Delete     variants
        5 = List       variants
        6 = See variant contents
        7 = Edit comments
        8 = End
          2

Enter: Name[:version] of variant to be modified = ad_bc2auhg

Modify definition? y
```

As you can see, the initial variant definition page of the newly created variant is the same as the one of variant “sch_bc1auhg”:

```
                          VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
xcscnfa      G   1

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
fscd         G   1  nc           G   1  p            G   1  rawem        G   1
tw           G   1  tt           G   1  t2t          G   1  tlk          NO BC
rfk          NO BC  sfk          NO BC  csfmr        NO BC  tend         G   1
rscmp        NO BC  shonf        G   1  ex7          NO BC  ex8          NO BC
zsc          NO BC  str          NO BC  wd           G   1  sat          G   1
shd          G   1  ant          NO BC  antl1        NO BC  struct1      NO BC
cheat2       NO BC  cardex       NO BC


HETEROSKEDASTICITY VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
tend         BC


AUTOCORRELATION STRUCTURE
   1         3         4        12
```

We will include two major modifications in our last experience. First, the dependent variable “xcanfa“ will be given its own Box-Cox parameter (place the cursor in the appopriate field and press “i” followed by “B”. And secondly, **quasi-dummy** variables “**rfk**” and “**ex7**” will be Box-Cox transformed. In the last section, we have insisted on the fact that Box-Cox transformed variables should be strictly positive. However, there is a way to weaken this constraint. A Box-Cox parameter may be associated with a positive variable which may contain zero values. The Box-Cox transformation of a zero value is defined to be zero. To include the null values in the model, an **associated dummy variable** is automatically added to the model.

Modify variant “ad_bc2auhg:0“ to obtain the following variant definition page without forgetting the new autocorrelation and heteroskedasticity structures:

```
                              VARIANT DEFINITION


DEPENDENT VARIABLE
|----------------|
xcanfa       BC

FIXED PART VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
fad           G   1  nc           G   1  p            G   1  rawem        G   1
tw            G   1  tt           G   1  t2t          G   1  tlk          NO BC
rfk          *G   1  sfk          NO BC  csfmr        NO BC  eanfp        G   1
rcutp         NO BC  shonf        G   1  ex7         *G   1  ex8          NO BC
ex9           NO BC  ex0          NO BC  ex1          NO BC  str          NO BC
za            NO BC  wd           G   1  sat          G   1  shd          G   1
ant           NO BC  antl1        NO BC  fir          NO BC

HETEROSKEDASTICITY VARIABLES
|----------------|  |----------------|  |----------------|  |----------------|
shonf        BC


AUTOCORRELATION STRUCTURE
   1        12
```

Note that an asterisk indicates that a quasi-dummy variable is Box-Cox transformed and that an associated dummy variable will be added to the model.

Do not forget any modification: future results could differ !! Continue the initialization of variant "ad_bc2auhg" giving the same answers as those used in the preceding example (as in variant sch_bc1auhg). At the end of the initialization process you will receive the message "Variant ad_bc2auhg:1 initialized".

In this last version, six additional parameters will be estimated during the log-likelihood function maximization process: one Box-Cox parameter for the dependent variable, one Box-Cox parameter for a group of variable in the fixed part of the model, one Box-Cox parameter for the heteroskedasticity variable, one heteroskedasticity coefficient, and two autocorrelation parameters.

If you ask for a list of the variants you will obtain the following table:

```
TRIO module: 2.10      Page:   1                    Date: 93/11/05                                    User: ....
Project: Tutorial Demonstration Database
Data type: level
-------------------------------------------------------------------------------------------------------------------------------


List of level variants:

   Variant name   Version   Status   Date of last modification
   ____________   _______   ______   _________________________

   ad_bc2auhg         0        I           93/11/05
   ad_bc2auhg         1        I           93/11/05
   sch_bc1auhg        0        I           93/11/05


   Created:  0    Initialized:  3    Estimated:  0   Total:  3
```

Note that variant ad_bc2auhg:0 has only been created as a transition to variant ad_bc2auhg and should be deleted. In particular, its name starts with the letters **ad** but its dependent variable is that used in variants with a name starting with the letters **sch**.

2.3 ESTIMATE VARIANTS

The variants to be estimated are first included in a waiting list managed by each TRIO estimation module. When the user starts the estimation procedure, TRIO goes through the list of variants ready to be estimated and estimates the variants in a "first in first out" order. The system is then unaccessible to the user until completion of all the requested estimations. On multitasking systems however, the user may continue to work on parts of the database or terminate the TRIO session.

2.3.1 Inserting a variant into the estimation waiting list

You must first build the list of variants to be estimated. The list may contain up to a hundred variants. Only initialized variants may be inserted in the list. Enter **module 3.10** and enter the "**sch_bc1auhg**" variant in the list using the "**Add**" option.

```
3.10  ESTIMATE LEVEL VARIANTS

Select: Operation on variants to be estimated
        1 = Add
        2 = List
        3 = Delete
        4 = Estimate
        5 = End
          1

Enter: Name[:version] of variant to be estimated = sch_bc1auhg

Enter: Name[:version] of variant to be estimated =
```

Look at the contents of the list using the "**List**" option.

```
              LIST OF VARIANTS TO BE ESTIMATED
              ________________________________



Present time = 14:35:11


No     Variant      Version   Status   Start date   Start time   End time
__   ___________   _______   ______   __________   __________   ________


  1   sch_bc1auhg        0       I
WAIT. Press any key to continue.
```

When finished just press "**Return**" and you will get back to the previous menu. Include variant "**ad_bc2auhg:1**" into the list of variants to be estimated.

```
              LIST OF VARIANTS TO BE ESTIMATED
              ________________________________


 Present time = 14:36:18


 No     Variant       Version   Status   Start date   Start time   End time
 __   _____________   _______   ______   __________   __________   ________


  1   sch_bc1auhg         0       I

  2   ad_bc2auhg          1       I
WAIT. Press any key to continue.
```

At the end of the estimation procedure, this list may be consulted to get information about computing times or abnormal terminations.

2.3.2 Estimating variants

Once all the variants to be estimated are in the list, initiate the process as follows: select the "**Estimate**" option. TRIO will estimate variants found in the list, ignoring any with status "E".

```
 3.10  ESTIMATE LEVEL VARIANTS

 Select: Operation on variants to be estimated
         1 = Add
         2 = List
         3 = Delete
         4 = Estimate
         5 = End
           4

 Select: Type of report
         1 = No    report
         2 = Short report
         3 = Long  report
           2

 Do you want to run your estimation(s) in "background" ? n

 Select: List device
         1 = Terminal
         2 = Print file
           1
```

TRIO reports on maximization steps during estimation (short or long report). For information concerning the contents of the maximization reports please refer to the Reference manual. For the purpose of our demonstration select the "**Short report**" option. TRIO also offers the possibility to run your estimation in background in a SUN station but we will not use this option at the present time.

An echo of the iterations performed is sent to the terminal. Thus you can follow the progression of the process. For information concerning the maximization algorithm please refer to the Model Types manual.

```
Select: List device
        1 = Terminal
        2 = Print file
          1
```

The report may be sent to the reports file or to the screen. Select "**Terminal**". After the estimation process, TRIO will give to the user the opportunity to print the estimation parameters.

```
Send to print file? y

Enter: Name of your output file (reports) = estimation
```

If you select the "**List**" option of **module 3.10** you will obtain the following list.

```
              LIST OF VARIANTS TO BE ESTIMATED
              ________________________________



Present time = 15:13:07


No     Variant      Version   Status   Start date   Start time   End time
__   ____________   _______   ______   __________   __________   ________

 1   sch_bc1auhg        0       E       93/11/05     15:09:07    15:10:23

 2   ad_bc2auhg         1       E       93/11/05     15:10:24    15:12:10
WAIT. Press any key to continue.
```

The status of the variants is now "E" for Estimated. If errors occur during the estimation process, an "aborted flag" signals to the user that the variant estimation has not been completed. In that case, the variant stays initialized (status "I"). The estimation time may vary from one computer facility to another. Even on the same computer, differences may be noticed, depending on the current load of the system. Times shown above were obtained on a SUN4 work station during the day.

2.3.3 Deleting a variant from the estimation waiting list

The waiting list keeps statistics on the estimated variants as long as you do not remove variants from the list. The "**Delete**" option removes the variants from the estimation list only. The variants themselves may only be deleted or modified in module 2.10. As an example, you may now try to remove the different estimated variants from the waiting list.

2.4 ANALYZE VARIANT RESULTS

TRIO provides the user with different tools to analyze variant results. This analysis can be carried out by means of lists, tables or graphs. For example correlation among the variables of a variant may be analyzed via a correlation matrix table. Comparisons between variants may be performed with a special table named "TABLEX". Graphically, variant results such as residuals, simulation forecasts or expected values of the dependent variable may be displayed. In this section, some of the analysis tools will be described.

2.4.1 Contents of a variant

As a first example we will show how we can quickly examine the contents of a particular variant. Enter **module 2.10** and select the "**See variant contents**" option. Enter "**sch_bc1auhg**" as the variant name and send the list to the terminal. You should get the following:

TRIO module: 2.10 Date: 93/01/27 User: .... Page: 2
Project: Tutorial Demonstration Database
Data type: level

CONTENTS OF VARIANT sch_bc1auhg VERSION 0

Variable name	Coefficient(I)	F/E	Student-t	B-C param.(I)	F/E	BCg	Coefficient(E)	Student-t	B-C param.(E)	Student-t
Dependent variable.										
xcscnfa	-	-		0.100000E+01	E	1	-		0.100463E+01	4.8026 0.0221
Fixed part variables.										
constant	-0.220379E+08	-	-2.2167				-0.244528E+08	-0.2801 -2.3634		
fscd	-0.193864E+08	-	-2.3093	0.100000E+01	E	1	-0.229702E+08	-0.2647 -2.7429	0.100463E+01	4.8026 0.0221
nc	-0.913518E+01	-	-1.3775	0.100000E+01	E	1	-0.103585E+02	-1.2653 -1.6911	0.100463E+01	4.8026 0.0221
p	-0.238817E+05	-	-0.7091	0.100000E+01	E	1	-0.383321E+04	-0.1101 -0.1167	0.100463E+01	4.8026 0.0221
rawem	0.356623E+05	-	1.4945	0.100000E+01	E	1	0.337014E+05	0.3721 1.0296	0.100463E+01	4.8026 0.0221
tw	-0.540029E+05	-	-0.3288	0.100000E+01	E	1	-0.272677E+06	-0.3432 -0.8701	0.100463E+01	4.8026 0.0221
tt	0.430551E+05	-	0.8012	0.100000E+01	E	1	-0.867892E+04	-0.0885 -0.0914	0.100463E+01	4.8026 0.0221
t2t	0.311024E+06	-	1.2739	0.100000E+01	E	1	0.270423E+06	0.3320 0.6653	0.100463E+01	4.8026 0.0221
tlk	0.104787E+04	-	3.1731				0.338926E+03	0.2752 0.6785		
rfk	-0.506567E+05	-	-0.8662				-0.489131E+05	-0.3192 -1.1489		
sfk	-0.359299E+05	-	-3.2514				-0.110086E+05	-0.3055 -1.1370		
csfmr	0.222120E+04	-	1.7917				0.170777E+04	0.2939 0.9550		
tend	0.179534E+05	-	2.1830	0.100000E+01	E	1	0.201038E+05	0.4477 1.7425	0.100463E+01	4.8026 0.0221
rscmp	0.432381E+01	-	11.8419				0.357708E+01	0.3240 6.4485		
shonf	0.207389E+01	-	3.7366	0.100000E+01	E	1	0.133834E+01	1.2434 1.5874	0.100463E+01	4.8026 0.0221
ex7	0.258583E+00	-	5.3809				0.214863E+00	0.3236 3.9104		
ex8	0.257051E+00	-	1.4151				0.225689E+00	0.3041 0.6291		

For each variable in the model you can find: the associated computed or estimated coefficient at the beginning and at the end of the estimation process. If a Box-Cox parameter is associated with a variable, the initial and final values are displayed. A special flag “F/E” indicates whether the parameters are fixed during estimation. The available values of that flag are “F” for fixed, “E” for estimated and “C” for computed parameter, the last option meaning that the user has no control on the way the parameter is evaluated by the system. Two Student-t statistics are given with each parameter. The reader can refer to the Model Types manual for an exact interpretation of these statistics.

Following the variables are the general parameters of the variant: the different period intervals used in the estimation and the maximization algorithm parameters. Finally, general statistics pertaining to the model are displayed for both initial and estimated values of the parameters.

```
Periods
_______

Intersection sample of variables X,Z   : [   1, 193]
Intersection sample of variables Y,X,Z : [   1, 193]
Available variant sample               : [   1, 193]
Variant sample                         : [   1, 193]
Estimation period                      : [  13, 181]
Periods at end of sample
used to evaluate elasticities          : [ 170, 181]
Prediction period                      : [ 182, 193]


Maximization algorithm parameters
_________________________________

Maximum number of iterations :            100
Number of iterations in run  :             14
Tolerance factor             :   0.100000E-04
Upper limit for Y            :   0.000000E+00
Weight for Hessian matrix    :   0.100000E+01


Statistics
__________

Initial values.

Log-likelihood value:               -0.233751E+04
Mean prob. of y to be at the limit:  0.206796E-08
Pseudo-(l)-Rsquare:                  0.906840E+00
Adjusted pseudo-(l)-Rsquare:         0.889782E+00
Pseudo-(e)-Rsquare:                  0.906840E+00
Adjusted pseudo-(e)-Rsquare:         0.889782E+00
Error variance:                      0.604442E+11


Final values (after estimation procedure).

Log-likelihood value:               -0.229734E+04   Ratio test:   0.803398E+02
Mean prob. of y to be at the limit:  0.390619E-07
Pseudo-(l)-Rsquare:                  0.942087E+00
Adjusted pseudo-(l)-Rsquare:         0.927931E+00
Pseudo-(e)-Rsquare:                  0.931826E+00
Adjusted pseudo-(e)-Rsquare:         0.915161E+00
Error variance:                      0.128532E+11   Student-t:    0.1649
                                                                  4.1757
```

2.4.2 Lists

Lists are the simplest tables to generate. Series of values such as variable contents, residuals, predicted values of the dependent variable, may be displayed in the form of columns of numbers. There is no limit to the number of columns to be displayed simultaneously but you can select the page width according to the printer and monitor in use.

But, before showing how it works, we will create two new variables from variant results. In order to create these variables from variant results we have to enter **module 1.10**, select "**Edit variables**". Then select the "**Add**" option and follow the instructions.

```
Select: Creation method
        1 = Copy a variable
        2 = Enter interactively
        3 = Take the difference of a variable
        4 = Define algebraic expression
        5 = Take from variant results
          5

Enter: Name of variable to be created = forec_y

Enter: Variant name [:version] to consider = sch_bc1auhg

Select: Values
        1 = type u residuals (regression errors)
        2 = type u residuals in % vs a variable
        3 = type v residuals (homoskedastic errors)
        4 = type v residuals in % vs a variable
        5 = type w residuals (final model errors)
        6 = type w residuals in % vs a variable
        7 = E(Y): expected values of Y
        8 = Probability of Y not to be at the limit
        9 = E(Y**): expected values of transformed Y
       10 = Standard error of Y
       11 = Maximum likelihood forecast of E(Y)
       12 = Simulation forecast of E(Y)
       13 = Standard error of forecast
          11
```

When finished, select "**No futher edition**". and **in the same way, create variable "std_err" using option 13** of the last select question.

```
Select: Creation method
        1 = Copy a variable
        2 = Enter interactively
        3 = Take the difference of a variable
        4 = Define algebraic expression
        5 = Take from variant results
          5

Enter: Name of variable to be created = std_err

Enter: Variant name [:version] to consider = sch_bc1auhg

Select: Values
        1 = type u residuals (regression errors)
        2 = type u residuals in % vs a variable
        3 = type v residuals (homoskedastic errors)
        4 = type v residuals in % vs a variable
        5 = type w residuals (final model errors)
        6 = type w residuals in % vs a variable
        7 = E(Y): expected values of Y
        8 = Probability of Y not to be at the limit
        9 = E(Y**): expected values of transformed Y
       10 = Standard error of Y
       11 = Maximum likelihood forecast of E(Y)
       12 = Simulation forecast of E(Y)
       13 = Standard error of forecast
          13
```

We can now create a list of the predicted school-children demand of model “sch_bc1auhg”. Enter **module 4.15** and follow the instructions.

```
4.15  LEVEL TABLES

Select: 1 = List     table
        2 = Cross    table
        3 = Tablex   table
        4 = End
          1

Select: List device
        1 = Terminal
        2 = Print file
          1

Enter: Table title (LEVEL LIST) = Non linear schoolchildren ridership
forecasts

Enter: First period to consider (    1) = 182

Enter: Last period to consider (  200) = 193

Define column number  1:

Select: Type of values
        1 = level variables
        2 = level variant results
          1

Enter: Algebraic expression = #t

Enter: Column description (#t ) = Period

Enter: Number of precision digits to consider (2) = 0

Another column? y

Define column number  2:

Select: Type of values
        1 = level variables
        2 = level variant results
          1
```

```
Enter: Algebraic expression = xcscnfa

Enter: Column description (STUDENT TRIPS) = Observed

Enter: Number of precision digits to consider (2) = 0

Another column? y

Define column number  3:

Select: Type of values
        1 = level variables
        2 = level variant results
          1

Enter: Algebraic expression = forec_y

Enter: Column description (forec_y) = Predicted

Enter: Number of precision digits to consider (2) = 0

Another column? y
```

```
 Define column number  4:

 Select: Type of values
         1 = level variables
         2 = level variant results
           1

 Enter: Algebraic expression = 100*abs((forec_y - xcscnfa) / xcscnfa)

 Enter: Column description (100 * abs((forec_y - xcscnfa) / xcscnfa) )
 = Relative error in %

 Enter: Number of precision digits to consider (2) = 2

 Another column? y

 Define column number  5:

 Select: Type of values
         1 = level variables
         2 = level variant results
           1

 Enter: Algebraic expression = (xcscnfa.gt.(forec_y+1.96*std_err)) -
(xcscnfa.lt.(forec_y-1.96*std_err))

 Enter: Column description ((xcscnfa .gt. (forec_y + 1.96 * std_err))
 - (xcscnfa .lt. (forec_y) = Out of 95% confidence interval

 Enter: Number of precision digits to consider (2) = 0

 Another column? n

 1 page(s) sent to the terminal.
```

The displayed list follows.

Non linear schoolchildren ridership forecasts

Period	Observed	Predicted	Relative error in %	Out of 95% confidence interval
182.	3852330.	3044450.	20.97	1.
183.	2921650.	2891982.	1.02	.
184.	4048180.	3479677.	14.04	.
185.	2591580.	2984457.	15.16	.
186.	3699270.	3214119.	13.11	.
187.	2221850.	2695921.	21.34	.
188.	1266040.	907198.	28.34	.
189.	1429810.	1171141.	18.09	.
190.	3869190.	3026325.	21.78	1.
191.	3895000.	3135731.	19.49	.
192.	3985440.	3184243.	20.10	.
193.	3361990.	2846759.	15.33	

2.4.3 Cross Tables

Cross tables are tables with two or more dimensions (matrices). Five types of cross tables are available in **module 4.15**: correlation matrices, covariance matrices, autocorrelation and partial autocorrelation functions for residuals, multicollinearity tables and prediction tables. As an example we will show how to create a covariance matrix table.

2.4.3.1 Covariance matrix

The following dialogue generates the asymptotic covariance of a subset of parameters of variant "sch_bc1auhg".

```
4.15  LEVEL TABLES

Select: 1 = List     table
        2 = Cross    table
        3 = Tablex   table
        4 = End
          2

Select: 1 = Correlation matrices
        2 = Covariance matrices
        3 = Autocorrelation analysis of residuals
        4 = Detection of multicollinearity
        5 = Prediction table
          2

Enter: Name[:version] of variant to consider = sch_bc1auhg

Enter: Table title ( sch_bc1auhg:0    COVARIANCE ANALYSIS ) =

Consider all parameters? n

Consider autocorrelation parameters? y

Consider heteroskedasticity coefficients? y

Consider Box-Cox parameters on X (and Y)? y

Consider heteroskedasticity Box-Cox parameters? y

Consider regression coefficients? n

Consider error variance? y

Computing covariance matrix ...

Select: List device
        1 = Terminal
        2 = Print file
          1
```

TRIO module: 4.15 Date: 93/01/27 User: .... Page: 19
Project: DEMONSTRATION DATABASE

sch_bc1auhg:0 COVARIANCE ANALYSIS

PARAMETERS	rho01	rho03	rho04	rho12	CH-tend	B-group 1	BH-tend	Error var.
rho01	.263948e-02	-.509209e-03	.968166e-04	-.461616e-03	.118952e-03	-.796346e-05	-.303977e-02	.786447e+07
rho03	-.509209e-03	.490992e-02	-.270299e-02	.932358e-04	-.605146e-03	-.422834e-05	.138191e-01	.131445e+09
rho04	.968166e-04	-.270299e-02	.447096e-02	-.564839e-03	.206751e-03	.469743e-06	-.490944e-02	-.258363e+08
rho12	-.461616e-03	.932358e-04	-.564839e-03	.272608e-02	-.324282e-03	-.582834e-05	.713374e-02	.933180e+08
CH-tend	.118952e-03	-.605146e-03	.206751e-03	-.324282e-03	.691080e-03	-.404874e-05	-.162935e-01	-.117975e+09
B-group 1	-.796346e-05	-.422834e-05	.469743e-06	-.582834e-05	-.404874e-05	.276249e-06	.990170e-04	.454813e+06
BH-tend	-.303977e-02	.138191e-01	-.490944e-02	.713374e-02	-.162935e-01	.990170e-04	.385201e+00	.271414e+10
Error var.	.786447e+07	.131445e+09	-.258363e+08	.933180e+08	-.117975e+09	.454813e+06	.271414e+10	.254945e+20

2.4.4 Variant comparisons with TABLEX

The TABLEX is another type of table available in TRIO, probably the most powerful one. It gives the user the possibility to perform direct variant comparisons. The generated tables can both be utilized as a rapid comparative evaluation-elimination of working variants or for final report presentation. Many aspects of each variant can be displayed simultaneously, enabling a direct column to column comparison. In this following section, we will define a TABLEX.

2.4.4.1 Reading a variant batch file

To really understand the interest of the TABLEX we will use a set of four new variants (sch_lin, sch_bc1, sch_bc1au and ad_bc1auhg) already estimated. This tutorial package includes a file named d211.in composed of a set of variants already defined, initialized and estimated for you. If you want, you could edit this file outside TRIO and examine how a batch file containing TRIO variants is constructed.

The following dialogue of **module 2.11** shows how to enter these four additional variants.

```
 2.11  READ/WRITE LEVEL VARIANTS

 Select: 1 = Read variants from ascii file
         2 = Write variants to ascii file
         3 = End
           1

 Enter: Name of your input file (d211.in) =  d211.in

 Echo the batch reading? n
 Variant "ad_bc1auhg" version   0 has been read.
 Variant "ad_bc1auhg" version   0 has been initialized.
 Variant "ad_bc1auhg" version   0 has been estimated.
 Variant "sch_bc1" version   0 has been read.
 Variant "sch_bc1" version   0 has been initialized.
 Variant "sch_bc1" version   0 has been estimated.
 Variant "sch_bc1au" version   0 has been read.
 Variant "sch_bc1au" version   0 has been initialized.
 Variant "sch_bc1au" version   0 has been estimated.
 Variant "sch_lin" version   0 has been read.
 Variant "sch_lin" version   0 has been initialized.
 Variant "sch_lin" version   0 has been estimated.
WAIT. Press any key to continue.
```

2.4.4.2 Group edition

When numerous variables are present in a database, the user may feel uncomfortable with the management and manipulation of variable names. TRIO provides the user with **groups of variables** that permit logical grouping of variables of the same type. For example, variables describing prices could be grouped together in a PRICES group. Like variables or variants, groups can be managed either interactively or by using batch input and output files. This implies that groups can be transported from one database to another.

To simplify the tutorial let us now define a general group that will include all the level variables of the database. Exit **module 2.11**, enter **module 1.10** and then select option "**2**" of the "Edit Level Variables" menu ("**Edit groups**"). Then select option "**1**" in the menu "**Operations on groups**".

```
Select: Operation on groups
        1 = Add
        2 = Modify
        3 = Delete
        4 = List
          1

Select: Creation method
        1 = Copy a group
        2 = Enter group interactively
          2

Enter: Name of group to be created = ORIGINAL

Enter: Group description = ALL LEVEL VARIABLES

Enter: Name of variable(s) to be added (max. 1000 characters)
= *

Default group? y

Group "ORIGINAL" now exists in database.
```

Since “ORIGINAL” will be our first group of reference, an order of 1 is assigned to it by answering “**yes**” to the “Default group?” question. For additional information concerning the “Group order” the user should refer to the Reference Manual.

Now, select again option “**2**” of the “Edit Level Variables” menu (“**Edit groups**”). Then select option “**2**” (Modify) in the menu “**Operations on groups**”. Answer “**ORIGINAL**” to the enter question “**Name of group to be modified =**” and press the return key following the question concerning the group order (1).

```
                                    ORIGINAL
ALL LEVEL VARIABLES                                          Order:  1
  variable  cat   variable  cat
ant          3  shonf        0
antl1        3  std_err      0
cardex       3  str          2
cheat2       3  struct1      2
complex      2  t2t          0
crowded      0  tend         0
csfmr        2  tlk          2
eanfp        0  tt           0
ex0          2  tw           0
ex1          2  wd           0
ex7          2  xcanfa       0
ex8          2  xcscnfa      0
ex9          2  za           2
fad          0  za_dummy     3
fir          3  zsc          2
forec_y      0
fscd         0
mixed        0
nc           0
one          0
p            0
rawem        0
rcutp        2
rfk          2
rscmp        2
sat          0
sfk          2
shd          0
```

You are now in the level variable group editor. Please note that a Help menu describing all the commands of the group editor is always available by pressing the “h” key. Again, the commands to move around the fields are: move up and down with commands “e” and “x”, left and right with commands “s” and “d”. To delete a variable, use command “y” and to replace a variable, use the insert command “i”.

Since the order in which the variables are listed is important, new commands are defined to add or move variables. The command “p” adds a new variable before the current one. The command “n” adds it after the current position. Pressing “>” (“<”) increases (decreases) the position of the variable in the group.

Now eliminate from the group all the variables created in the last section (complex, mixed, etc...) using the key “**y**”.

The numbers appearing besides each variable represents its **category** in the group. In our example, the variables should be given the following categories:

```
                                   ORIGINAL
ALL LEVEL VARIABLES                                          Order:  1
  variable  cat   variable  cat
ant          3  shonf        0
antl1        3  std_err      0
cardex       3  str          2
cheat2       3  struct1      2
complex      2  t2t          0
crowded      0  tend         0
csfmr        0  tlk          2
eanfp        0  tt           0
ex0          2  tw           0
ex1          2  wd           0
ex7          2  xcanfa       0
ex8          2  xcscnfa      0
ex9          2  za           2
fad          0  za_dummy     3
fir          3  zsc          2
forec_y      0
fscd         0
mixed        0
nc           0
one          0
p            0
rawem        0
rcutp        0
rfk          0
rscmp        0
sat          0
sfk          2
shd          0
```

The correct categories are assigned to the corresponding variables with the variable group editor command "**i**" followed by the category number, followed by "Return" or <ESC>. When all the modifications have been performed, quit the editor with the command "**k**" which saves the corrections.

The construction of the group named ORIGINAL served only as an example. Delete this group and read from the ASCII file the groups that will actually be used in the TABLEX. To delete the group, select "**Edit groups**" in menu 1.10, then select the option "**Delete**", and finally, enter the name "**ORIGINAL**". To read the groups from the ASCII file, call module 1.11 and select the option "**Read groups from ascii file**"; the file to be read is the default file g111.in.

2.4.4.3 TABLEX creation

Now we can proceed with the construction of a TABLEX. Exit **module 1.10**, enter again **module 4.15** and follow the instructions.

```
4.15  LEVEL TABLES

Select: 1 = List     table
        2 = Cross    table
        3 = Tablex   table
        4 = End
          3

Select: Printing format
        1 = normal
        2 = compressed
          2

Select: Page  width
        1 =  80 columns
        2 = 128 columns
        3 = 160 columns
          2

Select: List device
        1 = Terminal
        2 = Print file
          1

Do you want the default number of lines per page (  70 ) ? y

Enter: Table title (LEVEL VARIANT COMPARISON) =
```

If you know in advance that the resulting table is ready to be printed, you can send it directly to the print file without displaying it on the screen. For the moment answer "**1**" to the "List device" question.

The type of results to be displayed is then asked: enter the values "**1 2 14 15**"

```
Enter: Data to be listed

        1  =  Regression coefficients

        2  =  Elasticity at the mean for the estimation period
        3  =  Elasticity of E(y) for the estimation period
        4  =  Elasticity of STE(y) for the estimation period
        5  =  Elasticity at the mean for a selected period
        6  =  Elasticity of E(y) for a selected period
        7  =  Elasticity of STE(y) for a selected period

        8  =  Partial derivative at the mean for the estimation period
        9  =  Partial derivative of E(y) for the estimation period
       10  =  Partial derivative of STE(y) for the estimation period
       11  =  Partial derivative at the mean for a selected period
       12  =  Partial derivative of E(y) for a selected period
       13  =  Partial derivative of STE(y) for a selected period

       14  =  Unconditional T-statistic
       15  =  Conditional T-statistic

Enter with a space between the items : 1 2 14 15

RESULT TO BE LISTED :  1   2  14  15
```

Now enter the variants to be considered. You can enter variants that do not describe the same dependent variable.

```
VARIANTS :
----------
Enter: Name of   1 variant[:version] : ad_bc1auhg
Enter: Name of   2 variant[:version] : ad_bc2auhg
Enter: Name of   3 variant[:version] : sch_lin
Enter: Name of   4 variant[:version] : sch_bc1
Enter: Name of   5 variant[:version] : sch_bc1au
Enter: Name of   6 variant[:version] : sch_bc1auhg
Enter: Name of   7 variant[:version] :
```

You have now to choose the variables that you want to see in the TABLEX. This is always done by using groups of variables.

```
GROUPS :
--------
Do you want the default groups? y

Do you want only variables that appear in variant? y

Do you want a detailed listing of the Box-Cox transformations? n
```

The default groups having already been read in the g111.in file, you may answer “yes” to the question “Do you want the default groups?”.

The first portion of the generated table describes the results associated with the variables of the model. You can obtain two types of results other than values. When a dash (-) appears it signifies that the statistic asked was not calculated by TRIO, when a series of asterisks (*) appears it implies that the statistic was calculated but the value was too big to appear (these results are unusual).

```
TRIO module: 4.15      Page:   1                    Date: 93/11/05                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------

                              LEVEL VARIANT COMPARISON

=============================================================================================
I.  BETA                        TYPE =LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1
    ELASTICITY    S(y) (EP)  VARIANT =ad_bc1au ad_bc2au sch_lin  sch_bc1  sch_bc1a sch_bc1a
    (UNCOND. T-STATISTIC)    VERSION =    0        1        0        0        0        0
    (COND.   T-STATISTIC)   DEP.VAR. = xcanfa   xcanfa  xcscnfa  xcscnfa  xcscnfa  xcscnfa
=============================================================================================
------
PRICES
------

  REAL MUCTC             fscd                          -.19E+08 -.25E+04 -.16E+06 -.23E+08
  STUDENT TRANSIT FARE                                    -.376    -.292    -.419    -.410
                                                        (-2.31)   (-.28)   (-.25)   (-.26)
                                                        (-2.31)  (-1.10)  (-2.39)  (-2.74)
                                                                  LAM  1   LAM  1   LAM  1

  REAL MUCTC             fad          -.50E+07 -.56E+06
  ADULT TRANSIT FARE                    -.193    -.193
                                       (-.43)   (-.40)
                                      (-7.41)  (-7.81)
                                       LAM  1   LAM  1

  CONSUMER PRICE         p             .59E+04  .36E+03 -.24E+05 -.26E+03  .80E+03 -.38E+04
  INDEX, MONTREAL                       .111     .060   -1.081   -1.596     .577    -.166
                                       (.36)    (.22)   (-.71)   (-.44)    (.27)   (-.11)
                                       (.47)    (.26)   (-.71)   (-.74)    (.30)   (-.12)
                                       LAM  1   LAM  1            LAM  1   LAM  1   LAM  1

-------------------------
MOTOR VEHICLE - QUANTITY
-------------------------

  CARS IN MUCTC AREA     nc           -.18E+02 -.17E+01 -.91E+01 -.23E+01 -.60E+01 -.10E+02
                                        -.393    -.329    -.923    -.710   -1.138   -1.036
                                      (-1.63)   (-.47)  (-1.38)   (-.69)  (-1.31)  (-1.27)
                                      (-3.42)  (-2.47)  (-1.38)   (-.80)  (-1.66)  (-1.69)
                                       LAM  1   LAM  1            LAM  1   LAM  1   LAM  1
```

```
TRIO module: 4.15      Page:   6                    Date: 93/11/05                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------

=============================================================================================
I.  BETA                        TYPE =LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1
    ELASTICITY    S(y) (EP)  VARIANT =ad_bc1au ad_bc2au sch_lin  sch_bc1  sch_bc1a sch_bc1a
    (UNCOND. T-STATISTIC)    VERSION =    0        1        0        0        0        0
    (COND.   T-STATISTIC)   DEP.VAR. = xcanfa   xcanfa  xcscnfa  xcscnfa  xcscnfa  xcscnfa
=============================================================================================
------------------------
ASSOCIATED DUMMIES GROUP
------------------------

  RAINFALL PER           rfk                    -.20E+05
  DAY, DORVAL AIRPORT    ===                       -.035
                                                  (-.42)
                                                 (-2.66)

  EXPO 67                ex7                     .21E+05
                         ===                        .037
                                                   (.39)
                                                  (2.12)
```

Variables underlined once are quasi-dummy variables. Such variables may contain zero values but are not Boolean (0 or 1). Variables underlined twice are real dummy variables. Variables which are

Box-Cox transformed are denoted with the special symbol “LAM”. If the Box-Cox parameter is shared by several variables, the group number appears as well (see variable “wd”).

Note that the last part of the first section of the table contains a group called “ASSOCIATED DUMMIES GROUP”, which is generated automatically by the TABLEX table when variants contain variables that have null observations but are nevertheless transformed by a Box-Cox transformation. In addition, if a variable has been used to model heteroskedasticity, the last group considered in the first section of the table will automatically be the “HETEROSKEDASTICITY STRUCTURE” group because the various results shown in this first section of the TABLEX table can also be obtained for these variables.

The second section of the table contains results associated with the other parameters of the model.

```
================================================================================================
I.  BETA                          TYPE =LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1
    ELASTICITY    S(y) (EP)  VARIANT =ad_bc1au ad_bc2au sch_lin  sch_bc1  sch_bc1a sch_bc1a
    (UNCOND. T-STATISTIC)    VERSION =    0        1        0        0        0        0
    (COND.   T-STATISTIC)   DEP.VAR. = xcanfa   xcanfa  xcscnfa  xcscnfa  xcscnfa  xcscnfa
================================================================================================
----------------------------
HETEROSKEDASTICITY STRUCTURE
----------------------------

  DELTA COEFFICIENTS
  ------------------

  REAL RETAIL            shonf         .57E-01  .13E-01
  SALES INDEX                             .044     .045
                                        (.02)    (.02)
                                       (1.12)   (1.07)
                                          LAM      LAM

  SECONDARY              tend                                                        .10E-01
  SCHOOL REGISTRATION
```

```
================================================================================================
II. PARAMETERS                    TYPE =LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1
    (UNCOND. T-STATISTIC)     VARIANT =ad_bc1au ad_bc2au sch_lin  sch_bc1  sch_bc1a sch_bc1a
    (COND.   T-STATISTIC)     VERSION =    0        1        0        0        0        0
                             DEP.VAR. = xcanfa   xcanfa  xcscnfa  xcscnfa  xcscnfa  xcscnfa
================================================================================================
----------------------------
HETEROSKEDASTICITY STRUCTURE
----------------------------

  BOX-COX TRANSFORMATIONS: UNCOND: [T-STATISTIC=0] / [T-STATISTIC=1]
  ------------------------------------------------------------------

  LAMBDA(Z)              shonf             .190     .305
                                          [.05]    [.07]
                                         [-.23]   [-.16]

  LAMBDA(Z)              tend                                                         1.074
                                                                                     [1.04]
                                                                                      [.07]

------------------------------------------------------------------
BOX-COX TRANSFORMATIONS: UNCOND: [T-STATISTIC=0] / [T-STATISTIC=1]
------------------------------------------------------------------

  LAMBDA(Y)              xcanfa                .787
                                             [5.77]
                                            [-1.56]

  LAMBDA(Y) - GROUP  1   LAM  1             .917                        .511     .723    1.005
                                           [7.27]                      [2.67]   [3.27]   [4.80]
                                           [-.66]                     [-2.56]  [-1.25]    [.02]

  LAMBDA(X) - GROUP  1   LAM  1             .917     .917               .511     .723    1.005
                                           [7.27]   [5.26]             [2.67]   [3.27]   [4.80]
                                           [-.66]   [-.48]            [-2.56]  [-1.25]    [.02]
```

```
---------------
AUTOCORRELATION
---------------

  ORDER  1              RHO  1          .009     .015                      .138     .255
                                       (.21)    (.33)                    (1.78)   (2.89)
                                       (.21)    (.34)                    (1.80)   (2.93)

  ORDER  3              RHO  3                                            -.344    -.186
                                                                        (-3.60)  (-1.74)
                                                                        (-3.84)  (-1.80)

  ORDER  4              RHO  4                                            -.086    -.055
                                                                        (-1.06)   (-.57)
                                                                        (-1.10)   (-.58)

  ORDER 12              RHO 12          .922     .918                      .382     .471
                                     (21.64)  (20.44)                    (4.52)   (6.36)
                                     (21.78)  (20.61)                    (4.69)   (6.60)
```

The value of the estimated parameters is displayed together with the two t-tests. Square brackets are used for the t-tests related with the Box-Cox parameters to insist on the fact that the null hypotheses are different than those associated with the other parameters (for additional information please refer to the Reference manual).

The last portion of the table contains log-likelihood values, pseudo-(L)-R^2 and other information on the variant being analyzed.

```
TRIO module: 4.15      Page:   8                      Date: 93/11/05                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------------

================================================================================================
III.GENERAL STATISTICS          TYPE =LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1  LEVEL-1
                             VARIANT =ad_bc1au ad_bc2au sch_lin  sch_bc1  sch_bc1a sch_bc1a
                             VERSION =    0        1        0        0        0        0
                            DEP.VAR. = xcanfa   xcanfa  xcscnfa  xcscnfa  xcscnfa  xcscnfa
================================================================================================
LOG-LIKELIHOOD                        -2414.20 -2407.54 -2337.51 -2333.42 -2306.62 -2297.34

PSEUDO-R2 : - (E)                        .971     .973     .907     .909     .934     .932
            - (L)                        .971     .973     .907     .911     .935     .942
            - (E) ADJUSTED FOR D.F.      .964     .965     .890     .891     .919     .915
            - (L) ADJUSTED FOR D.F.      .964     .966     .890     .894     .921     .928

AVERAGE PROBABILITY (Y=LIMIT OBSERV.)    .000     .000     .000     .000     .000     .000

SAMPLE : - NUMBER OF OBSERVATIONS         169      169      169      169      169      169
         - FIRST OBSERVATION               13       13       13       13       13       13
         - LAST  OBSERVATION              181      181      181      181      181      181

NUMBER OF ESTIMATED PARAMETERS :
  - FIXED PART :
      . BETAS                             28       30       27       27       27       27
      . BOX-COX                            1        2        0        1        1        1
      . ASSOCIATED DUMMIES                 0        2        0        0        0        0
  - AUTOCORRELATION                        2        2        0        0        4        4
  - HETEROSKEDASTICITY :
      . DELTAS                             1        1        0        0        0        1
      . BOX-COX                            1        1        0        0        0        1
      . ASSOCIATED DUMMIES                 0        0        0        0        0        0
================================================================================================
```

The variants shown in this TABLEX table correspond to those found in Model Types version 1.0, chapter 7, Table 3 for the following cases: ad_bc1auhg corresponds to Adults 2.3; sch_bc1, sch_bc1au and sch_bc1auhg correspond to Schoolchildren 2.0, 2.1 and 2.3 (absolute values of the log likelihood differ by an arbitrary constant but differences between nested cases are indentical).

2.4.5 Graphic analysis

Using **module 4.10**, variant results may be analyzed and compared with level variables. Select the "**Horizontal line graph option**". Enter "**VARIANT ANALYSIS**" as a title. We will use two data sets: the dependent variable "xcscnfa" and its expected value obtained by the model.

```
Select: Graph type
        1 = Horizontal line graph
        2 = Vertical   line graph
        3 = Scatter         graph
        4 = Bar             graph
        5 = Pie             graph
          1

Enter: Graph title  = VARIANT ANALYSIS

Enter: Character size [1 - 6: default = 2] =

Enter: Data sets to consider (1) = 2

Scale Y values? n
```

Different output formats are available when more than one set of values is to be displayed. If a scaling of the Y values is chosen, a Z-score transformation is performed on the values of each curve. This is useful when comparing values of different range. When no scaling is requested, the curves are displayed in their originals values. Select "**no scale**".Then you will have to select which variant result you want to plot.

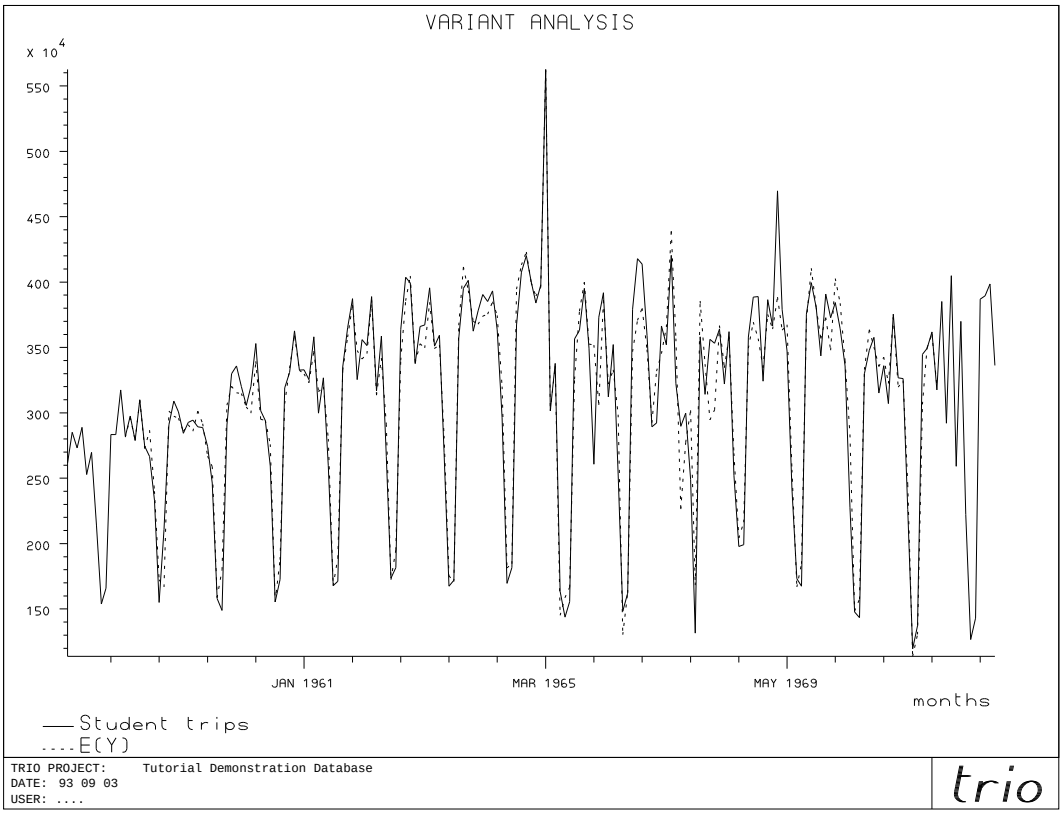
```
Enter: X axis description:(time) = months

Enter: Character size [1 - 6: default = 2] =

Do you wish to display labels for X values? y

Define Y axis values:

Select: Type of values
        1 = level variables
        2 = level variant results
          1

Enter: Algebraic expression = xcscnfa

Enter: Y axis description (STUDENT TRIPS) = Student trips

Enter: Character size [1 - 6: default = 2] =

Y values are defined for period    1 to   193

Enter: First period to consider =

Enter: Last period to consider =

Another Y variable? y

Define Y axis values:

Select: Type of values
        1 = level variables
        2 = level variant results
          2

Enter: Variant name [:version] to consider = sch_bc1auhg
```

```
Select: Values
        1 = type u residuals (regression errors)
        2 = type u residuals in % vs a variable
        3 = type v residuals (homoskedastic errors)
        4 = type v residuals in % vs a variable
        5 = type w residuals (final model errors)
        6 = type w residuals in % vs a variable
        7 = E(Y): expected values of Y
        8 = Probability of Y not to be at the limit
        9 = E(Y**): expected values of transformed Y
       10 = Standard error of Y
       11 = Maximum likelihood forecast of E(Y)
       12 = Simulation forecast of E(Y)
       13 = Standard error of forecast
          7
```

As already seen TRIO offers thirteen different variant results. Select the "**E(Y): expected values of Y**" option using estimated values of the parameters (initial values would produce the same output in this case). **Use the default values for the other answers**. Refer to the Model Types manual for more information concerning each of these choices.

```
Using estimated results? y

Enter: Y axis description (E(Y)) =

Enter: Character size [1 - 6: default = 2] =

Y values are defined for period   13 to   181

Enter: First period to consider =

Enter: Last period to consider =

Save in the temporary list of graphs? y

Plot? y
```

The resulting graph represents the observed (continuous line) and expected (dotted line) values of the dependent variable.



To exit from the graphic session press the key "**q**"

2.5 PERFORMING A BACKUP OF THE DATABASE

As a security measure it is always advised to do a regular backup of the variables, groups and variants included in a database. This option is also particularly useful when variants or variables have to be transferred from one database to another.

2.5.1 Variable backup

As an example we are going to perform a backup of all the variables of our database. To do this, exit **module 4.10**, enter in **module 1.11** and **select** the **"Write variables to ascii file"** (option **3**).

Answer "*****" to the question "Variable names to be written?" in order to save all the variables present in the database.

TRIO will then ask you "Do you want to write the descriptions and comments?". You should always answer "**yes**" to this question in order to save all the descriptions and comments entered during the TRIO session.

The next question asked is "If applicable, do you want algebraic expressions instead of values?". Again answer "**yes**" if you want to save the newly created algebraic expressions.

```
1.11  READ/WRITE LEVEL VARIABLES

Select: 1 = Read variables from ascii file
        2 = Read groups from ascii file
        3 = Write variables to ascii file
        4 = Write groups to ascii file
        5 = End
          3

Variable names to be written (max. 1000 characters)
= *

Do you want to write the descriptions and comments? y

If applicable, do you want algebraic expressions instead of values? y
```

TRIO will then ask you for the name of the output file:

```
Enter: Name of your output file (d111.out) = Montreal.in
```

When the "write" phase is over, **press <CR>**

2.5.2 Groups backup

Proceed in the same way to make a backup up of the groups created in the database. **Select** the "**Write groups to ascii file**" (option 4). Then TRIO will ask for the name of the output file:

```
Enter: Name of your output file (g111.out) = Montgrp.in
```

Answer "*****" to the question "Group names to be written" in order to save all the groups included in the database. TRIO will then ask you "Do you want to write the descriptions and the order?". You should always answer "**yes**" to this question and save all the descriptions and order number of the groups entered during the TRIO session.

When the "write" phase is over, you can **press <CR>** and exit **module 1.11** by typing the key "**q**" or by selecting option "**5**".

2.5.3 Variant backup

The following dialogue of **module 2.11** shows how to save all our level variants present in the database.

```
2.11  READ/WRITE LEVEL VARIANTS

Select: 1 = Read variants from ascii file
        2 = Write variants to ascii file
        3 = End
          2

Do you want to write particular variant versions? n

Enter: Name of variant(s) to be written = *
Do you want all versions ( last version if not)? y

 7 variant(s) to be written.

Do you want to write the comments? y


Enter: Name of your output file (d211.out) = level.out
```

All the variants have been written in an external ASCII file named **level.out** (output file). You can type this file on your screen to see what it looks like. In order to do this, you have to exit TRIO (unless you work with a multitask computer such as SUN). While out of TRIO you could also rename the batch output file **level.out**. TRIO gives you the opportunity to read any input file as long as it is constructed following the TRIO structure.

If after this short introduction to TRIO you have more detailed questions concerning some commands of TRIO you should consult the **Reference Manual** (or the **Model Types Manual** for the econometric and statistical questions you may have).

CHAPTER 3: PROBABILITY CLASS

3.1 PROBABILITY VARIABLES

Probability variables are defined on individuals. They are divided in three types: the individual choice, the explanatory and the alternative availability variables.

3.1.1 Creating new variables

As for level variables there are several ways of creating new probability variables. In this section we will present in particular how to create an availability variable and how to create a variable from an availability variable.

3.1.1.1 Reading probability batch input file

But first we will, in the same way as for level variables, read in batch mode a certain number of probability variables needed in this Tutorial (again the user will find all the information he needs to create the batch input files in the Reference manual).

To do so, enter **module 1.31** and select the "**Read variables from ascii file**" option. **Do not echo** the batch reading in order to shorten the process and press **Return** to the question "Enter: Name of your input file (d131.in) =" to select the default name (the default probability input batch file of the tutorial database is named **d131.in**).

```
 1.31  READ/WRITE PROBABILITY VARIABLES

 Select: 1 = Read variables from ascii file
         2 = Read groups and alternatives from ascii file
         3 = Write variables to ascii file
         4 = Write groups and alternatives to ascii file
         5 = End
           1

 Enter: Name of your input file (d131.in) =

 Echo the batch reading? n
 Variable "tdv1" read.
  . . . .
 Variable "choice" read.
 Variable "dispon" read.
WAIT. Press any key to continue.
```

At the end of the Read operation (at the WAIT. Press any key to continue. prompt) press "**Return**".

3.1.1.2 Creating an availability variable

We will now proceed with the creation of a new availability variable. First exit **module 1.31** by selecting "**End**" and enter **module 1.30**. Then, select "**Edit variables**" followed by the "**Add**" option. Select "**Enter interactively**", name the new variable to be created "**dispnew**" and then select the variable type: "**Alternative availability**".

```
1.30  EDIT PROBABILITY VARIABLES

Select: 1 = Edit variables
        2 = Edit groups
        3 = Edit alternatives
        4 = End
          1

Select: Operation on variables
        1 = Add
        2 = Modify
        3 = Delete
        4 = List
        5 = Edit comments
          1

Select: Creation method
        1 = Copy a variable
        2 = Enter interactively
        3 = Define algebraic expression
        4 = Take from variant results
        5 = Take from availability variable
          2

Enter: Name of variable to be created = dispnew

Select : Variable type
        1 = Individual choice
        2 = Explanatory
        3 = Alternative availability
          3
```

Enter "**1**" and "**622**" as first and last individuals respectively.

```
Enter: First individual = 1

Enter: Last individual = 622
```

An availability variable is defined on all possible alternatives which may be inserted in a variant. The alternative number is used to define the availability of that alternative for each individual. If alternative one, for example, is defined as the bus mode, every use of alternative one in this database will refer to this mode.

By default all alternatives are available to all individuals. In order to get a reduced choice set, we must enter the number of the alternatives we wish to work with. (enter "**1**" and "**4**"). TRIO will then ask you to enter the individuals for which the alternative is not available and those for which the value is a missing data, considering the alternative as available for the rest of the individuals.

```
 Enter: Alternatives to be considered (max 10 per line)
 =1 4
 =

 Number of available alternative(s) selected:  2

 Enter: First individual [last individual [step]]
        for which alternative   1 is not available
 = 1 10
 =

 Enter: First individual [last individual [step]]
        for which the availability of alternative   1
        is unknown (missing data)
 =

 Enter: First individual [last individual [step]]
        for which alternative   4 is not available
 = 1
 =

 Enter: First individual [last individual [step]]
        for which the availability of alternative   4
        is unknown (missing data)
 =
```

Following the creation, the comment page may be edited. The comment menu is displayed every time a new variable is created. Select the "**Edit a comment**" option and press "**Return**" at the enter variable name question.

```
 Select: 1 = Edit  a comment
         2 = Copy  a comment
         3 = Print a comment
         4 = List  a comment
         5 = Change table type
         6 = List   table contents
         7 = No further edition
           1

 Enter: Variable name (dispnew) =
```

```
line:   3   col:  11   insert  ON    left:   1   page: 1   EMPTY

    type: probability_              variable name: dispnew_____
    user: ....__________            date: 93/11/08

description1: ____________________________________________________________

description2: ____________________________________________________________
comments:
_________!_________!_________!_________!_________!_________!_________!_________!
```

Go to the field "description1" of this page. Insert a short description such as "**New availability modified for alternative one**". To do this, press "**i**" to get in insert mode and enter the string, ending with **<Esc> or <CR>**. After entering the first description, we will define a part of this string as a block and copy it on the second description line and on the next comments page.

First, let us define the block containing the string to be copied: position the cursor at the beginning of the description and press "**bb**" (**b**lock **b**egin). Move the cursor to the end of the string "New availability" and press "**be**" (**b**lock **e**nd). The selected block is in reverse mode or highlighted on your screen. This block is selected to be copied using the command "**bw**"(**b**lock **w**rite). Move the cursor at the beginning of the next field "description2" and press "**br**" (**b**lock **r**ead).

Please note that it is not necessary to use the buffer if a block is to be copied or moved **within the same field**: the commands "**bc**"(**b**lock **c**opy) and "**bm**" (**b**lock **m**ove) will perform these tasks more simply than "**br**". More details on the commands are given in appendix A and in the Reference Manual.

```
line:   8   col:  15   insert  ON    left:  49   page: 1   L EXT:  17

    type: probability_               variable name: dispnew_____
    user: ...._____________          date: 93/11/08

description1: New availability modified for alternative one_____________________

description2: New availability ____________________________________________________
comments:
_________!_________!_________!_________!_________!_________!_________!_________!
```

Commands "**ml**" and "**mr**" modify the margins of the comment field. Insert a long text in the comment field. When inserting text in the comment field, a wrap around is automatically done as the right margin is reached. For example:

```
comments:
_________!_________!_________!_________!_________!_________!_________!_________!


The variable dispnew has been created in order to explain the modifications of
an availability variable. It is also used for the text editor example.
This variable should not appear in any variant since it does not represent avail
ability of a sample.
```

The two portions of text have been entered without a **<Carriage Return>**. A <CR> has been inserted between the two portions of text. You must press **<Esc>** to end the insertion. Now press "**mr**" to modify the right margin. The message "**Column number**" appears on the margin line. One could enter a column number or use the cursor position by pressing the "**Return**" key (enter **50**). The text is automatically reformatted as follows.

```
comments:
_________!_________!_________!_________!_________!


The variable dispnew has been created in order to
explain the modifications of
an availability variable. It is also used for the
text editor example.
This variable should not appear in any variant sin
ce it does not represent availability of a sample.
```

To get the next page of comments or to get a new one, press "**n**". When no consecutive page is present the user must, by answering "**y**", specify if a new page is really wanted. The pages are independent of each other. You must leave the current page before editing a new one. In our case do not save it. The second page has only a comment field. We will now recover the text we saved in the register by pressing "br". The text is inserted as follows:

```
line:   5   col:   1   insert  OFF   left: 795   page: 2   L EXT:  17
comments:
_________!_________!_________!_________!_________!

New availability
```

Return to the previous page by pressing "**p**" without saving the current page. To quit editing without saving the current page use "**q**".

3.1.1.3 Algebraic expressions

New choice and new explanatory variables may be defined by using algebraic expressions with the same syntax as previously seen for level variables. However there are some exceptions. In particular if the new variable to be created will be defined as a choice variable, the resulting vector will be truncated to integer values. Also, negatives values and values exceeding the maximum number of alternatives will be set to zero and considered as missing data. For additional information concerning the probability choice variables the user should refer to the Reference manual.

3.1.1.4 Using an availability variable

In the following example we will create a new variable from the values of an availability variable. "**Add**" a new explanatory variable and select "**Take from availability variable**".

```
Select: Creation method
        1 = Copy a variable
        2 = Enter interactively
        3 = Define algebraic expression
        4 = Take from variant results
        5 = Take from availability variable
          5

Enter: Name of variable to be created = avail1

This new variable will be kept and treated as an explanatory variable

Enter: Associated alternative number = 1

Enter: Variable name to consider = dispon

Enter: Alternative to consider = 1
```

The values of "avail1" will be: 0, if the alternative one is not available, 1 if it is available and -1 if the availability is unknown.

3.1.2 Modifying existing variables

Different aspects of a variable may have to be updated. The names, the first and last individuals, the associated alternative and the values themselves. The values of a variable may have to be modified for various reasons: erroneous values may have been detected; new values may become available; missing data may be filled by predicted values. Some of these changes affect only a small number of observations while others may alter almost every individual for which the variable is defined. Interactive modifications and batch modifications are used in each one of these cases.

3.1.2.1 Interactive modification

Modification of the values of a **probability or choice variable** is performed in the same way as for level variables but it differs for **availability variables** (the variable may be modified by alternative or by individual). When modifying a variable that is used in a variant, TRIO displays the variants in which the variable is included. If modifications of values are made, the associated variant results are not valid any more and are lost: such variants have their status reset to "created". We will now modify the values of the availability variable "dispnew" created earlier. Enter **module 1.30** and select the "**Edit variables**" option.

```
Select: Operation on variables
        1 = Add
        2 = Modify
        3 = Delete
        4 = List
        5 = Edit comments
          2

Enter: Name of variable(s) to be modified (max. 1000 characters)
= dispnew
=

Variable to be modified: dispnew

Select: 1 = Modify value(s)
        2 = Modify name
        3 = Modify first and last individuals
        4 = Modify unit and first observation date
        5 = Modify  alternative
        6 = No modification
          1
```

Each alternative has to be considered if, for some individuals, some values have to be modified. Because the first alternative is not available for all individuals we will modify the availability of alternative one for individuals 3 to 6.

Instead of entering values, as in batch mode, a list of codes are used when performing the modifications interactively. The available codes are: **na** for not available, **av** for available and **md** for missing data (unknown).

Enter "**1**" for the alternative to be considered, and **3 to 6** for the individuals. Change the values as follows: for individuals 3 and 4, alternative 1 becomes available, for individual 5 the availability is unknown (missing data) and for individual 6 keep the same value.

```
Enter: Alternatives to be considered (max 10 per line)
= 1
=

Number of available alternative(s) selected:  1

Enter: First individual [last individual [step]] for alternative    1
= 3 6

indiv.   present value     new value     saved value
------   -------------     ---------     -----------

    3          na          =    av
                                          av
    4          na          =    av
                                          av
    5          na          =    md
                                          md
    6          na          =    <CR>
                                          na
```

3.1.3 Deleting variables

Deletion is performed in the same way as for level variables. If there are related variants to the variables to be deleted, their name will be displayed and a warning will be issued. If deletion occurs, those variants will also be deleted. As an exercise "**Delete**" variables "dispnew" and "avail1" that will not be used any more.

3.2 DEFINE AND INITIALIZE PROBABILITY VARIANTS

In this section we will show how to create a probability variant of the type **Box-Cox Logit** (for additional information concerning this type of model the user should refer to the Reference and Model Types manuals).

3.2.1 Create a probability variant

Enter **module 2.30**, select "**Create variants**" and then "**Enter interactively**". The first defined variant will be non linear and will use all the individuals included in the database. We will name this variant "**tot1lambda**".

We must first specify which alternatives are to be included in the variant. The Santiago data is defined over nine alternatives, so we will enter all the alternatives, **omitting a constant term only for alternative number 2**.

```
Select: Creation method
        1 = Copy a variant
        2 = Enter interactively
          2

Enter: Name of variant to be created = tot1lambda

Enter: Alternative number for equation  1 = 1

Introduce constant? y

Enter: Alternative number for equation  2 = 2

Introduce constant? n

Enter: Alternative number for equation  3 = 3

Introduce constant? y

Enter: Alternative number for equation  4 = 4

Introduce constant? y

Enter: Alternative number for equation  5 = 5

Introduce constant? y

Enter: Alternative number for equation  6 = 6

Introduce constant? y

Enter: Alternative number for equation  7 = 7

Introduce constant? y

Enter: Alternative number for equation  8 = 8

Introduce constant? y

Enter: Alternative number for equation  9 = 9

Introduce constant? y

Enter: Alternative number for equation 10 =
```

TRIO quits the selection of an alternative as soon as no alternative number is entered and then presents the full page variant editor (note that the commands associated with this editor are the same as the ones used in the level variant editor).

Enter the name of the dependent variable under the heading “Dependent”; in this case the variable should be “**choice**”. Next, under the field “Availability”, you should enter the variable “**dispon**” and please note that no “Weight” variable will be included in this variant since the sample is not a choice-based sample.

Now, move the cursor to the first utility function (besides "**1 C**") using the usual keys "**x**", "**d**", "**s**" and "**e**", and insert the variable names in order to get the following model structure. Remember that in order to enter a particular variable into the variant structure, you have to press the key "**i**" followed by the name of the variable (for example "**autlic**") and then press "**Return**"

Also, to enter the Box-Cox transformation (**G 1** for group 1), you must place the cursor on the appropriate field **over** the column of variables to be transformed and enter, in this case, "**i1**" followed by "**Return**"

```
line  16col 66              VARIANT DEFINITION

         DEPENDENT     AVAILABILITY  WEIGHT
         choice        dispon

BOX COX  NO BC         G   1         G   1         G   1         G   1
ALTER.
 1 C     autlic                     tcam1       * tdv1          cosing1
 2                                  tcam2       * tdv2          cosing2     *
 3 C                    tesp3       tcam3         tdv3          cosing3     *
 4 C                    tesp4       tcam4         tdv4          cosing4
 5 C                    tesp5       tcam5         tdv5          cosing5
 6 C     autlic         tesp6       tcam6         tdv6          cosing6
 7 C                    tesp7       tcam7         tdv7          cosing7
 8 C                    tesp8       tcam8         tdv8          cosing8
 9 C                    tesp9       tcam9         tdv9          cosing9
```

Move the cursor one field to the right of variable "cosing1" and insert the last column. The screen has scrolled to the left showing:

```
line  14col 66              VARIANT DEFINITION

         DEPENDENT     AVAILABILITY  WEIGHT
         choice        dispon

BOX COX  G   1         G   1         G   1         G   1         NO BC
ALTER.
 1 C                    tcam1       * tdv1          cosing1
 2                      tcam2       * tdv2          cosing2     * sexo
 3 C     tesp3          tcam3         tdv3          cosing3     * sexo
 4 C     tesp4          tcam4         tdv4          cosing4
 5 C     tesp5          tcam5         tdv5          cosing5
 6 C     tesp6          tcam6         tdv6          cosing6
 7 C     tesp7          tcam7         tdv7          cosing7       sexo
 8 C     tesp8          tcam8         tdv8          cosing8
 9 C     tesp9          tcam9         tdv9          cosing9
```

When the definition of the variant is completed, save and quit the new variant using the key "**k**".

3.2.2 Initialize a probability variant

As for level variants, you must initialize the probability variants to be able to estimate them later. So answer "**yes**" to the question "**Initialize the variant?**".

```
Initialize variant? y
Enter: First estimation individual (    1) =

Enter: Last estimation individual (  622) =
Validating data ...
Validating availability ...
Validating values ...
Validating model ...

Do you want to modify initial values of constants? n

Do you want to modify initial values of parameters? n
```

Initial values may be provided to start minimization elsewhere than at the default values of the parameters. Since no previous values of the parameters are available, **answer "no"** to the questions "Do you want to modify initial values of constants?" and "Do you want to modify initial values of parameters?". Complete the initialization, by setting the "Maximum number of iterations in likelihood function maximisation algorithm" to **100**, and using the default values for the "Tolerance factor in likelihood function maximisation algorithm" and "Weight for the Hessian",. Finally **do not scale** the data.

```
Enter: Maximum number of iterations in likelihood
       function maximization algorithm ( 100) = 100

Enter: Tolerance factor in likelihood function
       maximization algorithm (.10E-04) =

Enter: Weight for the Hessian (.10E+01) =

Scaling? n

Save initialization? y

Initializing variant ...   The initialization in process
computes a set of initial constants needed for the R-SQUARED.


Select: Type of report
        1 = No    report
        2 = Short report
        3 = Long  report
          2

Select: List device
        1 = Terminal
        2 = Print file
          1
```

A short description of the maximization process progress is displayed. At the end of the initialization answer "**yes**" to the question "**Do you want to keep these new constants as starting point for estimation?**" in order to be able to start the estimation process from the "starting point" just found.

At the end of the initialization exit the module by selecting "**No futher edition**".

3.3 ESTIMATE VARIANTS

Once the probability variant "**tot1lambda**" has been successfuly initialized you can proceed to its estimation.

3.3.1 Estimation waiting list

For the estimation of probability variants we use the same process ("**Add, Delete, List**") as for level variants. Enter **module 3.30** and select the option "**Add**". Then enter the name of the only variant initialized, "**tot1lambda**", in the estimation list.

```
3.30  ESTIMATE PROBABILITY VARIANTS

Select: Operation on variants to be estimated
        1 = Add
        2 = List
        3 = Delete
        4 = Estimate
        5 = End
          1

Enter: Name[:version] of variant to be estimated = tot1lambda

Enter: Name[:version] of variant to be estimated =
```

```
3.30  ESTIMATE PROBABILITY VARIANTS

Select: Operation on variants to be estimated
        1 = Add
        2 = List
        3 = Delete
        4 = Estimate
        5 = End
          2
```

The list of variants to be estimated should look like:

```
               LIST OF VARIANTS TO BE ESTIMATED
               ________________________________


Present time = 11:19:00

No     Variant      Version   Status   Start date   Start time   End time
__   ____________   _______   ______   __________   __________   ________

 1   tot1lambda         0       I
```

3.3.2 Run estimation procedure

To estimate the variant select "**Estimate**" from the "**Operation on variants to be estimated**" menu and follow the instructions.

```
 3.30  ESTIMATE PROBABILITY VARIANTS

 Select: Operation on variants to be estimated
         1 = Add
         2 = List
         3 = Delete
         4 = Estimate
         5 = End
           4
```

```
 Select: Type of report
         1 = No    report
         2 = Short report
         3 = Long  report
           2

 Do you want to run your estimation(s) in "background" ? n

 Select: List device
         1 = Terminal
         2 = Print file
           1

 ESTIMATION OF VARIANT : tot1lambda   VERSION :   0


 MAXIMIZATION PROCEDURE

 Stopping criteria:   nstep:       100
                      epsi:  0.100E-04
....
```

TRIO gives you the opportunity to run your estimations in background as previously explained but for our purposes we will answer "**no**" to the question about the background estimation. The maximization

steps of the objective function is displayed on the screen. Remember that all the variants remain in the list, until you remove them.

3.4 ANALYZING VARIANT RESULTS

Now that variant “tot1lambda” has been estimated we can proceed to analyze it. Enter **module 4.35**. As an example we will present, in this section, the correlation and the prediction success tables associated with variant “tot1lambda”.

3.4.1 Correlation matrices

Correlation analysis is performed on each group of independent variables specified by the selected alternatives, a matrix being displayed for each selected alternative.

Correlation analysis of the first two alternatives of variant “tot1lambda”, using original variables, is obtained as follows:

```
4.35  PROBABILITY TABLES

Select: 1 = List     table
        2 = Cross    table
        3 = Prediction success table
        4 = Tablex   table
        5 = End
          2

Select: 1 = Correlation matrices
        2 = Detection of multicollinearity
        3 = Simulated prediction
          1

Enter: Name[:version] of variant to consider = tot1lambda

Select: Type of variables
        1 = Original
        2 = Transformed using initial parameters
        3 = Transformed using estimated parameters
          1

Enter: Alternatives to be considered (max 10 per line)
=1 2
=

Number of available alternative(s) selected:  2

Enter: Table title ( tot1lambda:0     CORRELATION AMONG ORIGINAL
INDEPENDENT VARIABLES) =

Select: List device
        1 = Terminal
        2 = Print file
          1

   2 page(s) sent to the terminal.
```

The following tables will be send to your screen:

1) For alternative 1:

```
TRIO module: 4.35                    Date: 93/02/15                    User: ....                    Page:   56
Project: DEMONSTRATION DATABASE
-------------------------------------------------------------------------------------------------------------------------------

 tot1lambda:0     CORRELATION AMONG ORIGINAL INDEPENDENT VARIABLES

ALTERNATIVE  1:

            autlic    tcam1     tdv1  cosing1   tcam1*

autlic      1.0000   -.0593    .0707   -.0936   -.0409

tcam1       -.0593   1.0000    .0137    .2455    .3501

tdv1         .0707    .0137   1.0000    .1754    .0154

cosing1     -.0936    .2455    .1754   1.0000    .1799

tcam1*      -.0409    .3501    .0154    .1799   1.0000
```

2) For alternative 2 (use the key "**n**", next page in order to obtain this following page):

```
TRIO module: 4.35                    Date: 93/02/15                    User: ....                    Page:   57
 Project: DEMONSTRATION DATABASE
 -------------------------------------------------------------------------------------------------------------------------------

    tot1lambda:0     CORRELATION AMONG ORIGINAL INDEPENDENT VARIABLES

   ALTERNATIVE  2:

                tcam2      tdv2   cosing2      sexo    tcam2*  cosing2*

 tcam2         1.0000     .0712     .0510     .0673     .3624    -.0962

 tdv2           .0712    1.0000     .3215     .0687     .0439     .1050

 cosing2        .0510     .3215    1.0000    -.0755     .0011     .4854

 sexo           .0673     .0687    -.0755    1.0000    -.0249     .2485

 tcam2*         .3624     .0439     .0011    -.0249    1.0000    -.0879

 cosing2*      -.0962     .1050     .4854     .2485    -.0879    1.0000
```

Press "**q**" to exit the correlation tables and answer "**no**" to the question "**Send to print file?**"

3.4.2 Prediction success table

The predictive capability of the model can be measured by a set of prediction tests computed in a prediction success table. Let us proceed by showing an example of such a table. Select the "**Prediction success table**" option.

```
4.35  PROBABILITY TABLES

Select: 1 = List     table
        2 = Cross    table
        3 = Prediction success table
        4 = Tablex   table
        5 = End
          3

Enter: Name[:version] of variant to be considered = tot1lambda

Using estimated results? y
Computing prediction success table of variant tot1lambda version 0 ..
```

Select the terminal as the listing device and the following table will appear on the screen:

```
TRIO module: 4.35                    Date: 93/02/15                  User: ....                    Page:   58
Project: Tutorial Demonstration DAatabase
Data type: probability
-------------------------------------------------------------------------------------------------------------------

Prediction success table of variant tot1lambda version    0

alternative                  1           2           3           4           5           6           7           8

  1                    46.2913      5.4141      4.9383      8.5888     11.5637     23.6032      4.1594      2.7166
  2                     5.3074      8.9083      2.8706      2.2962      6.6160      4.1518      2.3201      1.0736
  3                     5.4821      2.2863      5.6500      0.7485      9.7097      6.6885      2.9030      2.3821
  4                     6.6614      1.5066      2.4005     88.9041      7.5639      1.2013      0.4629      0.3022
  5                    10.8548      6.1346     10.5838      7.9250     27.4834      8.9393      5.6022      4.5624
  6                    22.5095      3.6364      6.3095      1.2578     11.9630     28.8973      5.7158      4.1896
  7                     6.2740      3.1383      2.0247      0.0000      4.9611      6.0179      3.2647      1.6192
  8                     2.5591      1.3939      2.1036      0.0000      4.6486      4.4091      1.7409      2.0102
  9                     9.0605      4.5814      6.1189      0.2795     13.4905     11.0917      5.8309      4.1440
pred.count            115.0000     37.0000     43.0000    110.0000     98.0000     95.0000     32.0000     23.0000
pred.share              0.1849      0.0595      0.0691      0.1768      0.1576      0.1527      0.0514      0.0370
prop.succ.predict.      0.4025      0.2408      0.1314      0.8082      0.2804      0.3042      0.1020      0.0874
success index           0.2176      0.1813      0.0623      0.6314      0.1229      0.1514      0.0506      0.0504
prop.err.pred.shar.     0.0000      0.0000      0.0000      0.0000      0.0000      0.0000      0.0000      0.0000
```

For additional information concerning the contents of this type of table, the user should refer to the Reference manual. To exit the prediction success table, press the key "**q**" and answer "**no**" to the question "**Send to print file?**".

3.4.3 Variant comparisons with TABLEX

The probability TABLEX follows the same design as for level variants. The output is modified to accommodate the multiple alternatives dimension of variants: results are listed by alternative. For each variable appearing in the utility functions, twelve results are available (some of them are vectors of values instead of a single value). Since the available results are different for this type of model, the appearance of the table differs slightly from the format used for level variant. However the same column to column display is still used.

3.4.3.1 Reading a set of variants from an ASCII file

We will now introduce in our database a group of 5 others variants included in the variant input batch file **d231.in** of the Tutorial database. These variants will be created, initialized and estimated by TRIO during the batch reading process.

The new variants will have the same variables present in "**tot1lambda**". Only the individuals used for estimation and the Box-Cox transformations will differ. The names of the variants having a linear utility will end by "**lin**". If one Box-Cox transformation is used, the name will end by "**1lambda**" and, if three Box-Cox transformations are used, it will end by "**3lambda**". The first part of the name will refer to the number of individuals used during the estimation. If all individuals are included in the estimation the name will begin by "**tot**" for "**total**", otherwise it will begin by "**red**" for "**reduced**".

Santiago observations are reordered in such a way that observations containing zeros are in the first 152 individuals. For this reason the "redlin" variants will be defined on individuals 153 to 622 only.

Enter **module 2.31** and select "**Read variants from ascii file**" and then press "**Return**" to accept the default file name d231.in. **To accelerate the process do not echo the batch reading**. After the reading process is terminated press "**q**" twice and you will exit this module.

3.4.3.2 Edit probability groups and alternatives

As for the level class models, prior to creating a TABLEX we need to define the groups of variables we want to appear in the probability TABLEX. In this next section we will show how to create these groups and also how to associate labels with the alternatives in order to facilitate the reading of the TABLEX output.

3.4.3.2.1 Edit groups

Groups of probability variables are defined and manipulated in the same way as groups of level variables. Each variable of a group is again characterized by its own category. As an exercice we will create three different groups containing variables of a same type. Measures of time are included in the first group, measures of cost in the second and socioeconomic variables in the third.

To do so enter **module 1.30**, select the "**Edit groups**" option and then the "**Add**" option.

```
1.30  EDIT PROBABILITY VARIABLES

Select: 1 = Edit variables
        2 = Edit groups
        3 = Edit alternatives
        4 = End
          2

Select: Operation on groups
        1 = Add
        2 = Modify
        3 = Delete
        4 = List
          1

Select: Creation method
        1 = Copy a group
        2 = Enter interactively
          2

Enter: Name of group to be created = TIME

Enter: Group description = LEVEL OF SERVICE (TIME)

Enter: Name of variable(s) to be added (max. 1000 characters)
= tcam*
= tdv*
= tesp*
=

Default group? y

Group "TIME" now exists in database.
```

Variable “tcam[1-9]” stands for walking time for modes 1 to 9 in minutes. Variable “tdv[1-9]” stands for in-vehicle travel time in minutes. Variable “tesp[3-9]” stands for total waiting time in minutes.

In the same way create the second and the third group containing the following variables.

```
Select: Creation method
        1 = Copy a group
        2 = Enter interactively
          2

Enter: Name of group to be created = COST

Enter: Group description = LEVEL OF SERVICE (COST)

Enter: Name of variable(s) to be added (max. 1000 characters)
= cosing*
=

Default group? y

Group "COST" now exists in database.


...

Enter: Name of group to be created = SOCIO

Enter: Group description = SOCIOECONOMIC

Enter: Name of variable(s) to be added (max. 1000 characters)
= sexo
= autlic
=

Default group? y

Group "SOCIO" now exists in database.
```

Variable “cosing[1-9]” corresponds to the fare of the mode divided by the net wage. “Sexo” is a dummy variable taking value 1 for males and 0 for females and “autlic” represents the number of cars divided by number of driver licenses in a family.

Now to see the composition of the groups select the “**Edit groups**“ option and make a **detailed list** of the groups. You should get the following output.

```
Group: TIME          Default order:   1
Description: LEVEL OF SERVICE (TIME)
tcam1           tcam2           tcam3           tcam4
tcam5           tcam6           tcam7           tcam8
tcam9           tdv1            tdv2            tdv3
tdv4            tdv5            tdv6            tdv7
tdv8            tdv9            tesp3           tesp4
tesp5           tesp6           tesp7           tesp8
tesp9
WAIT. Press any key to continue.
```

```
 Group: COST         Default order:   2
 Description: LEVEL OF SERVICE (COST)
 cosing1         cosing2         cosing3         cosing4
 cosing5         cosing6         cosing7         cosing8
 cosing9
WAIT. Press any key to continue.
```

```
 Group: SOCIO        Default order:   3
 Description: SOCIOECONOMIC
 autlic          sexo
WAIT. Press any key to continue.
```

Remember that during the TABLEX construction TRIO will allow you to select which group will be displayed and in which order.

3.4.3.2.2 Edit alternatives

Up to now alternatives have only been referred to as numbers. In order to label these alternatives, a name and a description may be associated with each one of them, even though the name of the alternative is not used by TRIO to identify it. You should **always refer to an alternative by its number**. The real use of the description label is that it will appear on the left part of the TABLEX providing a more informative output during the TABLEX analysis.

Select the "**Edit alternatives**" and "**Modify**" option. Then enter alternative numbers **1 to 9** as the alternatives to be modified.

```
1.30  EDIT PROBABILITY VARIABLES

Select: 1 = Edit variables
        2 = Edit groups
        3 = Edit alternatives
        4 = End
          3

Select: Operation on names and descriptions
        1 = Modify
        2 = List
        3 = Initialize
          1

List of alternatives

 1
 2
 3
 4
 5
 6
 7
 8
 9

Enter: Alternatives to be considered (max 10 per line)
= 1 2 3 4 5 6 7 8 9
=
```

For each of the selected alternatives, **enter the name and the description as described below**.

```
Number of available alternative(s) selected:  9

Modify alternative  1:

Enter: Name (            ) = CARD

Enter: New description (max. 60 characters)
 (                                                                  )
= CAR DRIVER

Modify alternative  2:

Enter: Name (            ) = CARP

Enter: New description (max. 60 characters)
 (                                                                  )
= CAR PASSENGER

Modify alternative  3:

Enter: Name (            ) = STAXI

Enter: New description (max. 60 characters)
 (                                                                  )
= SHARED TAXI

Modify alternative  4:

Enter: Name (            ) = TRAINMET

Enter: New description (max. 60 characters)
 (                                                                  )
= TRAIN/METRO

Modify alternative  5:

Enter: Name (            ) = BUS

Enter: New description (max. 60 characters)
 (                                                                  )
= BUS
```

```
 Modify alternative  6:

 Enter: Name (            ) = PANDRIDE

 Enter: New description (max. 60 characters)
  (                                                                  )
 = PARK AND RIDE

 Modify alternative  7:

 Enter: Name (            ) = KANDRIDE

 Enter: New description (max. 60 characters)
  (                                                                  )
 = KISS AND RIDE

 Modify alternative  8:

 Enter: Name (            ) = STAXMET

 Enter: New description (max. 60 characters)
  (                                                                  )
 = SHARED TAXI-METRO

 Modify alternative  9:

 Enter: Name (            ) = BUSMET

 Enter: New description (max. 60 characters)
  (                                                                  )
 = BUS-METRO

 List of alternatives

  1 CARD         CAR DRIVER
  2 CARP         CAR PASSENGER
  3 STAXI        SHARED TAXI
  4 TRAINMET     TRAIN/METRO
  5 BUS          BUS
  6 PANDRIDE     PARK AND RIDE
  7 KANDRIDE     KISS AND RIDE
  8 STAXMET      SHARED TAXI-METRO
  9 BUSMET       BUS-METRO
WAIT. Press any key to continue.
```

If you wish to reset an alternative description use the "**Initialize**" option. This will erase the selected alternative descriptions and you will be able to enter a new description.

The groups file g131.in exists in the database; you could delete the groups that were just created and read the predefined groups.

3.4.3.3 The TABLEX construction

We will now produce a full TABLEX with the six experiments present in the database. To do so enter **module 4.35** and select the **TABLEX** option.

```
Select: Printing format
        1 = normal
        2 = compressed
          2

Select: Page  width
        1 =  80 columns
        2 = 128 columns
        3 = 160 columns
          2

Select: List device
        1 = Terminal
        2 = Print file
          1

Do you want the default number of lines per page (  70 ) ? y

Enter: Table title (PROBABILITY VARIANT COMPARISON) =
```

Then a list of the available statistics for this type of Tabex will be displyed. **Select each of them**.

```
Enter: Data to be listed

        1  =  Regression coefficients
        2  =  Gradient

        3  =  Elasticities  at the sample mean of X (direct/cross)
        4  =  Change in P(i) relative to percent change in X(j,k)
              at the sample mean of X (direct/cross)
        5  =  Derivative of P(i) with respect to X(j,k) in
              original units at the sample mean of X
              (direct/cross)
        6  =  Marginal rate substitution at the sample mean of X

        7  =  Weighted aggregate elasticities (direct/cross)
        8  =  Weighted aggregate change in P(i)
              relative to percent change in X(j,k) (direct/cross)
        9  =  Weighted aggregate derivative of P(i)
              with respect to X(j,k) in original units (direct/cross)
       10  =  Weighted aggregate marginal rate substitution

       11  =  Unconditional T-statistic
       12  =  Conditional T-statistic

Enter with a space between the items : 1 2 3 4 5 6 7 8 9 10 11 12
```

In order to reduce the TABLEX creation time **do not select cross results**.

```
For ELASTICITIES AT MEAN
Do you want the list of cross values? n

For CHANGE IN P(I) AT MEAN
Do you want the list of cross values? n

For DERIV. OF P(I) AT MEAN
Do you want the list of cross values? n

For W.A. ELASTICITIES
Do you want the list of cross values? n

For W.A. CHANGE IN P(I)
Do you want the list of cross values? n

For W.A. DERIVATIVE P(I)
Do you want the list of cross values? n

RESULT TO BE LISTED :  1   2   3   4   5   6   7   8   9  10  11  12
```

Then enter the variant names, the alternatives to be displayed and the variables to be used for the evaluation of the “Marginal Rate of Substitution” as follow:

```
VARIANTS :
----------
Enter: Name of   1 variant[:version] : redlin
Enter: Name of   2 variant[:version] : totlin
Enter: Name of   3 variant[:version] : red1lambda
Enter: Name of   4 variant[:version] : tot1lambda
Enter: Name of   5 variant[:version] : red3lambda
Enter: Name of   6 variant[:version] : tot3lambda
Enter: Name of   7 variant[:version] :

Alternatives to be considered :  1  2  3  4  5  6  7  8  9

Do you want all of these alternatives ? y
```

```
MARGINAL RATE SUBSTITUTION :
----------------------------
Enter:  Variable name for alternative  1 : cosing1
Enter:  Variable name for alternative  2 : cosing2
Enter:  Variable name for alternative  3 : cosing3
Enter:  Variable name for alternative  4 : cosing4
Enter:  Variable name for alternative  5 : cosing5
Enter:  Variable name for alternative  6 : cosing6
Enter:  Variable name for alternative  7 : cosing7
Enter:  Variable name for alternative  8 : cosing8
Enter:  Variable name for alternative  9 : cosing9

GROUPS :
--------
Do you want the default groups? y

Do you want only variables that appear in variant? y

Do you want a detailed listing of the Box-Cox transformations? n
```

The default groups defined earlier will be used, and only the variables that appear in at least one alternative will be displayed. After the initialization, which may be quite long, a trace of the construction is displayed.

Some tables showing some of the results obtained are given below.

```
TRIO module: 4.35      Page:   1                    Date: 94/06/15                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------

                            PROBABILITY VARIANT COMPARISON

=================================================================================================
I.  BETA                        TYPE = LOGIT    LOGIT    LOGIT    LOGIT    LOGIT    LOGIT
    GRADIENT                 VARIANT = redlin   totlin  red1lamb tot1lamb red3lamb tot3lamb
    ELASTICITIES AT MEAN     VERSION =    1       1        1        0        1        1
    CHANGE IN P(I) AT MEAN  DEP.VAR. = choice   choice   choice   choice   choice   choice
    DERIV. OF P(I) AT MEAN
    MRS AT MEAN:cosing1
    W.A. ELASTICITIES
    W.A. CHANGE IN P(I)
    W.A. DERIVATIVE P(I)
    W.A. MRS:   cosing1
    (UNCOND. T-STATISTIC)
    (COND.   T-STATISTIC)

=================================================================================================
```

The order of the twelve statistics is displayed on the left part. The variable used for the marginal rate of substitution is indicated on the right of the string “MRS AT MEAN:” and “W.A. MRS :”. T-statistics always appear at the end of the direct results. If cross results had been selected, they would appear after the t-statistics. The alternative number, name and description are displayed next.

You can obtain two types of results other than values. When a dash (-) appears it signifies that the statistic requested was not calculated by TRIO, and when a series of asterisks (*) appears it implies that the statistic requested was calculated but the value was too large to be printed.

```
ALTERNATIVE   1
 ==============
 ------------------------
 LEVEL OF SERVICE (TIME)
 ------------------------

   WALKING TIME           tcam1        -.13E+00 -.16E+00 -.44E+00 -.54E+00 -.39E+00 -.78E+00
   (CAR DRIVER)                        -.12E-03 -.59E-04 -.10E-03 -.29E-04 -.35E-04 -.44E-04
                          (1)             -.728    -.837    -.786    -.886    -.802    -.828
                                          -.041    -.042    -.041    -.043    -.041    -.039
                                       -.71E-02 -.77E-02 -.71E-02 -.78E-02 -.70E-02 -.72E-02
                                        .56E+01  .69E+01  .39E+01  .65E+01  .40E+01  .60E+01
                                          -.446    -.458    -.468    -.451    -.469    -.445
                                          -.116    -.122    -.133    -.126    -.132    -.133
                                       -.27E-01 -.33E-01 -.53E-01 -.54E-01 -.48E-01 -.74E-01
                                        .56E+01  .69E+01  .40E+01  .52E+01  .30E+01  .75E+01
                                        (-5.66)  (-7.82)  (-3.20)  (-3.30)  (-1.98)  (-3.44)
                                        (-5.66)  (-7.82)  (-6.05)  (-7.86)  (-6.09)  (-7.95)
                                           (GE)     (GE) L  1(GE) L  1(GE) L  1(GE) L  1(GE)

 ...
```

For each variable, besides the description and the name, the twelve statistics are displayed on different lines for each variant. The number of the current alternative is displayed under the variable name. The status of the variable is indicated on the last results line beside the Box-Cox transformation parameter number. The status is indicated by “GE” for generic variables, and by “SP” for specific variables.

The results are compared columnwise within each alternative. A special “ASSOCIATED DUMMIES GROUP” will automatically appear at the end of the groups for each alternative for which the utility function required such dummy variables because an explanatory variable that contained null observations was transformed by a Box-Cox transformation. We have introduced dummy variables in variants “tot1lambda” and “tot3lambda”

```
------------------------
ASSOCIATED DUMMIES GROUP
------------------------

                        tcam1                                -.14E+01          -.13E+01
                        =====                                 .38E-04          -.16E-05
                        (1)                                    -1.296            -1.210
                                                                 -.062             -.057
                                                              -.62E-01          -.57E-01
                                                               .52E+02           .48E+02
                                                                 -.864             -.798
                                                                 -.264             -.243
                                                              -.26E+00          -.24E+00
                                                               .25E+02           .24E+02
                                                               (-2.79)           (-2.60)
                                                               (-2.84)           (-2.63)
                                                                  (GE)              (GE)

-------------------------------------------------------------------------------------------

  REGRESSION CONSTANT    CONSTANT       .23E+00  .38E-01  .75E+00  .56E+00  .82E+00  .57E+00
                                        .33E-05  .28E-05 -.21E-04  .31E-04  .24E-04  .13E-04
                         (1)              -        -        -        -        -        -
                                          -        -        -        -        -        -
                                          -        -        -        -        -        -
                                          -        -        -        -        -        -
                                          -        -        -        -        -        -
                                          -        -        -        -        -        -
                                          -        -        -        -        -        -
                                          -        -        -        -        -        -
                                        (.45)    (.09)    (1.37)    (.90)   (1.50)    (.88)
                                        (.45)    (.09)    (1.39)    (.91)   (1.51)    (.94)
```

It is important to note the status of associated dummies. If a variable is specific, the associated dummy will also be specific. Otherwise, if two or more generic variables have associated dummies, they will be generic. If only one of the generic variables has an associated dummy, it will be specific.

If a constant term has been included for the current alternative, the available results will be displayed. At the end of the table, general statistics are available. Note that "_" indicates that a result is not applicable.

```
TRIO module: 4.35      Page:  25                    Date: 94/06/11                                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------------

================================================================================================
II. PARAMETERS                    TYPE = LOGIT    LOGIT    LOGIT    LOGIT    LOGIT    LOGIT
    UNCOND.[T-STATISTIC=0]    VARIANT = redlin   totlin  red1lamb tot1lamb red3lamb tot3lamb
    UNCOND.[T-STATISTIC=1]    VERSION =    1        1        1        0        1        1
                             DEP.VAR. = choice   choice   choice   choice   choice   choice
================================================================================================
  LAMBDA(X)              tdv8                                                    1.561    1.630
                         (8)                                                    [2.43]   [2.74]
                                                                                 [.87]   [1.06]

  LAMBDA(X)              cosing9                                                  .229     .353
                         (9)                                                    [1.21]   [1.46]
                                                                               [-4.08]  [-2.68]

  LAMBDA(X)              tdv9                                                    1.561    1.630
                         (9)                                                    [2.43]   [2.74]
                                                                                 [.87]   [1.06]

  LAMBDA(X) - GROUP  1   LAM  1                          .364     .326     .445     .062
                                                       [2.12]   [1.89]   [1.55]    [.30]
                                                      [-3.70]  [-3.91]  [-1.93]  [-4.53]

================================================================================================
III.GENERAL STATISTICS            TYPE = LOGIT    LOGIT    LOGIT    LOGIT    LOGIT    LOGIT
                              VARIANT = redlin   totlin  red1lamb tot1lamb red3lamb tot3lamb
                              VERSION =    1        1        1        0        1        1
                             DEP.VAR. = choice   choice   choice   choice   choice   choice
================================================================================================
LOG-LIKELIHOOD -FINAL VALUE           -657.273 -855.956 -652.382 -842.647 -650.006 -837.931
               -INITIAL VALUE         -657.273 -855.956 -652.382 -924.687 -650.006 -837.931
               -WITH CONSTANTS ONLY   -703.633 -924.687 -703.633 -924.687 -703.633 -924.687
               -RATIO TEST              92.720  137.462  102.503  164.079  107.253  173.512

RHO-SQUARED                             .066     .074     .073     .089     .076     .094
RHO-SQUARED BAR -AKAIKE                 .046     .059     .052     .070     .052     .073
                -HOROWITZ               .056     .067     .062     .080     .064     .084
                -HENSHER AND JOHNSON    .061     .070     .067     .084     .070     .089

PER CENT RIGHT                        46.383   47.106   45.319   48.232   43.830   49.518

SAMPLE -NUMBER OF ALTERNATIVES             9        9        9        9        9        9
       -NUMBER OF OBSERVATIONS           470      622      470      622      470      622
       -FIRST OBSERVATION                153        1      153        1      153        1
       -LAST  OBSERVATION                622      622      622      622      622      622
       -AVAILABLE OBSERVATIONS IN VT     470      622      470      622      470      622
       -AVAIL OBS IN   1 CARD            389      420      389      420      389      420
       -AVAIL OBS IN   2 CARP            318      466      318      466      318      466
       -AVAIL OBS IN   3 STAXI           419      560      419      560      419      560
       -AVAIL OBS IN   4 TRAINMET        109      141      109      141      109      141
       -AVAIL OBS IN   5 BUS             464      613      464      613      464      613
       -AVAIL OBS IN   6 PANDRIDE        331      375      331      375      331      375
       -AVAIL OBS IN   7 KANDRIDE        308      421      308      421      308      421
       -AVAIL OBS IN   8 STAXMET         326      435      326      435      326      435
       -AVAIL OBS IN   9 BUSMET          343      456      343      456      343      456

NUMBER OF ESTIMATED PARAMETERS
       -BETAS .VARIABLES                   6        6        6        6        6        6
              .CONSTANTS                   8        8        8        8        8        8
              .ASSOCIATED DUMMIES          0        0        0        2        0        2
       -BOX-COX TRANSFORMATIONS            0        0        1        1        3        3

================================================================================================
```

For more information concerning these statistics, please refer to the Reference manual and the Model Type manual. Note that the variants totlin:1, redlin:1 and red1lambda:1 correspond to the results found in Model Types, version 1.0, chapter 11, Table 5 (columns 0, 1 and 2) and that the variant red3lambda:1 corresponds to the results found in Model Types, version 1.0, chapter 11, Footnote 7.

To end displaying the TABLEX press "**q**". Answer "**yes**" to the question, "Send to print file?", and name the output file so that at the end of this session you will be able to examine the contents of the report file in detail.

3.4.4 Graphic analysis

Probability variables may also be analyzed, as for level variables, using Horizontal, Scatter, Bar and Pie graphs. In this section, as an example we will show how to create a Pie graph using the results of variant "totlin".

3.4.4.1 Pie graph

In order to visualize the relative distribution of values, the Pie graph is usually used. The percentage associated with each part is generated either with the values themselves or by using the percentage inside an interval.

By using the choice probability for one individual in a variant result, one gets percentages without having to define intervals. The number of parts will be equal to the number of alternatives in the variant.

Let us proceed with the example. Select **module 4.30** and then "**Pie graph**" (option **5**).

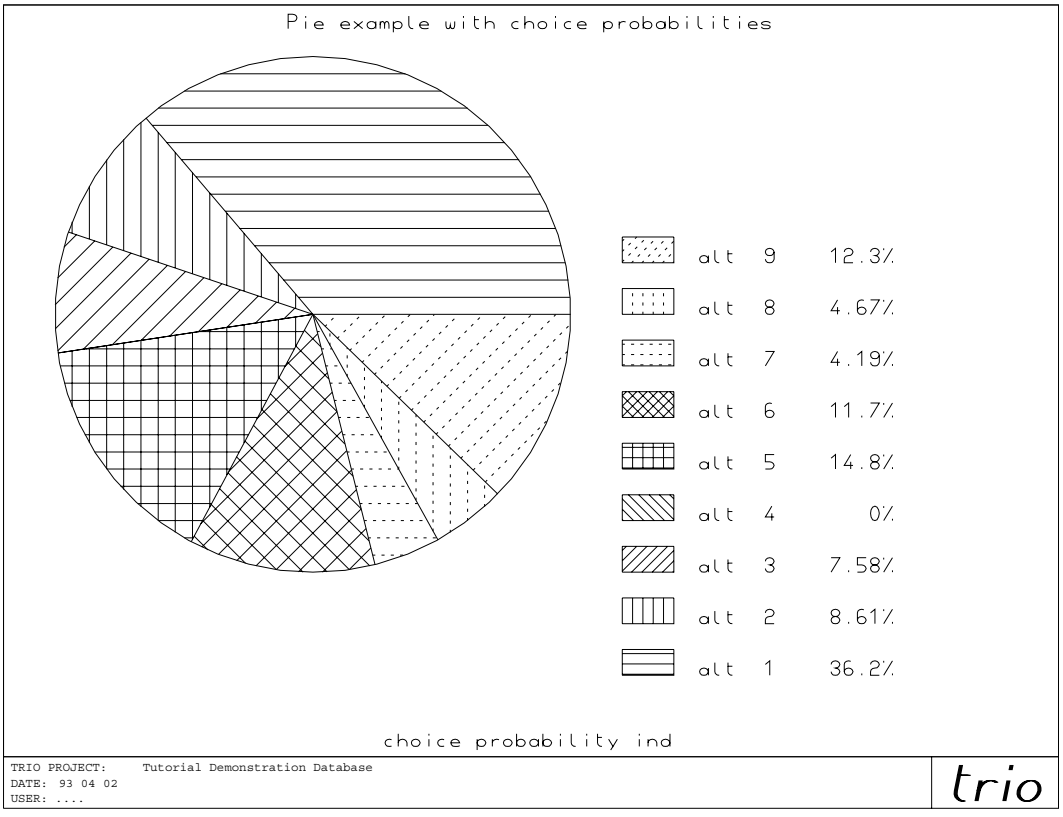
```
Select: Graph type
        1 = Horizontal line graph
        2 = Vertical   line graph
        3 = Scatter         graph
        4 = Bar             graph
        5 = Pie             graph
          5

Enter: Graph title  = Pie example with choice probabilities

Enter: Character size [1 - 6: default = 2] =

Select: Type of values
        1 = probability variables
        2 = probability availability
        3 = probability variant results
          3

Enter: Variant name [:version] to consider = totlin

Select: Values
        1 = Residuals for one individual(regression errors)
        2 = Residuals for one alternative(regression errors)
        3 = Choice probability for one individual
        4 = Choice probability for one alternative
        5 = Simulated prediction for one individual
          3

Using estimated results? y

Computing PNI ...
```

Choice probability values have to be recomputed each time they are selected for a different variant.

```
Enter: Individual number = 6

Enter: Description (choice probability ind) =

Enter: Character size [1 - 6: default = 2] =

Enter: Description of the part (alt  1) =

Enter: Character size [1 - 6: default = 2] =

Another subset? y
```

By answering yes, a subset for each alternative is obtained.. **Complete the construction for the nine alternatives by answering yes to the question " Another subset?"** .

```
....

Another subset? y

Enter: Description of the part (alt  9) =

Enter: Character size [1 - 6: default = 2] =

Save in the temporary list of graphs? y

Plot? y
```



3.5 PERFORMING A BACKUP OF THE DATABASE

As already mentioned it is always advised to do a regular backup of the variables, groups and variants included in a database. This option is also particularly useful when variants or variables have to be transferred from one database to another.

3.5.1 Variable backup

Again we are going to perform a backup of the variables of our database, this time of the probability variables. To do this, exit **module 4.35**, enter in **module 1.31** and **select** the **"Write variables to ascii file"** (option **3**).

Answer "*****" to the question "Variable names to be written?" in order to save all the variables present in the database.

TRIO will then ask you "Do you want to write the descriptions and comments?". You should always answer "**yes**" to this question in order to save all the descriptions and comments entered during the TRIO session.

The next question asked is "If applicable, do you want algebraic expressions instead of values?". Again answer "**yes**" if you want to save the newly created algebraic expressions.

```
1.31  READ/WRITE PROBABILITY VARIABLES

Select: 1 = Read variables from ascii file
        2 = Read groups and alternatives from ascii file
        3 = Write variables to ascii file
        4 = Write groups and alternatives to ascii file
        5 = End
          3

Variable names to be written (max. 1000 characters)
= *

Do you want to write the descriptions and comments? y

If applicable, do you want algebraic expressions instead of values? y
```

TRIO will then ask you for the name of the output file:

```
Enter: Name of your output file (d131.out) = Santiago.in
```

When the "write" phase is over, **press Return**.

3.5.2 Groups backup

Proceed in the same way to make a backup up of the groups created in the database. **Select** the "**Write groups and alternatives to ascii file**" (option 4). Then TRIO will ask for the name of the output file:

```
Enter: Name of your output file (g131.out) = Santgrp.in
```

Answer "*****" to the question "Group names to be written" in order to save all the groups included in the database. TRIO will then ask you "Do you want to write the descriptions and the order?". You should always answer "**yes**" to this question and save all the descriptions and order number of the groups entered

during the TRIO session. Then answer "**yes**" to the questions "Do you want to write the alternatives?" and "Do you want to write the description?" to save the alternatives used in the variants and their description.

When the "write" phase is over (at the "WAIT. Press any key to continue." prompt), you can **press Return** and exit **module 1.31** by typing the key "**q**" or by selecting option "**5**".

3.5.3 Variant backup

Now to do a backup of the variants present in the database enter **module 2.31**. The following dialogue of **module 2.31** shows how to save all the probability variants (created, initialized and estimated).

```
2.31  READ/WRITE PROBABILITY VARIANTS

Select: 1 = Read variants from ascii file
        2 = Write variants to ascii file
        3 = End
          2

Do you want to write particular variant versions? n

Enter: Name of variant(s) to be written = *
Do you want all versions ( last version if not)? y

 6 variant(s) to be written.

Do you want to write the comments? y


Enter: Name of your output file (d231.out) = prob.out
```

All the variants will be written to an external ASCII file named "prob.out" (output file). You can later view this file on your screen to see what it looks like.

To end the probability session press "**Return**" at the "**WAIT. Press any key to continue.**" prompt and then "**3**" or "**q**". Press "**Return**" once more and the main "Variable Editor" will be displayed.

If after this short introduction to TRIO probability models you have additional questions concerning some of TRIO's commands, you should always consult the **Reference Manual** (or the **Model Types Manual**).

CHAPTER 4: SHARE CLASS

Due to the number of different models available in the "share class", in this chapter we will mainly focus on the creation of variants associated with some of the share class models.

4.1 SHARE VARIABLES

Share variables are defined on observations and they are divided in two groups: the market shares and the explanatory variables. The creation rules that apply to share variables, groups and alternatives are very similar to those of probability variables. This is why we will only show in this section how to read these variables, groups and alternatives in batch mode.

4.1.1 Creation of share variables

As for level and probability variables, there are several ways of creating new share variables. Because we have already given the main instructions for creating new variables in the previous chapters, we will only show in this section how to read the share variables batch input file needed in this Tutorial. The user will find all the information he needs to create the batch input files in the Reference manual.

To do so, enter **module 1.21** and select the "**Read variables from ascii file**" option. **Do not echo** the batch reading in order to shorten the process and press **Return** to the question "Enter: Name of your input file (d121.in) =" to select the default name (the default share input batch file of the tutorial database is named **d121.in**).

```
1.21  READ/WRITE SHARE VARIABLES

Select: 1 = Read variables from ascii file
        2 = Read groups and alternatives from ascii file
        3 = Write variables to ascii file
        4 = Write groups and alternatives to ascii file
        5 = End
          1

Enter: Name of your input file (d121.in) =

Echo the batch reading? n
```

At the end of the Read operation (at the WAIT. Press any key to continue. prompt) press "**Return**".

4.1.2 Creation of share groups and alternatives

As for the level and probability class models, we need to define the groups of variables we want to display in the share TABLEX. In particular, in this next section we will show how to enter these groups and alternatives into the database in batch mode.

Groups of share variables are defined and manipulated in the same way as groups of level variables. Each variable of a group is again characterized by its own category.

To do so enter **module 1.21** and then select the "**Read groups and alternatives from ascii file**" option.

```
 1.21  READ/WRITE SHARE VARIABLES

 Select: 1 = Read variables from ascii file
         2 = Read groups and alternatives from ascii file
         3 = Write variables to ascii file
         4 = Write groups and alternatives to ascii file
         5 = End
           2

 Enter: Name of your input file (g121.in) =

 Echo the batch reading? n
 Group "N" read.
 Group "S" read.
 Alternatives read.
WAIT. Press any key to continue.
```

To accelerate the process "**do not echo the batch reading**" and at the **WAIT. Press any key to continue.** prompt press "**Return**". Once the groups and alternatives have been read, exit **module 1.21** by selecting option 5 ("**End**").

To be familiar with the share variables just entered in the database, let us make a list of the descriptions of the variables included in the two groups. To do so **enter module 1.20**, select "**Edit variables**" and press **Return**. Then select "**List**" (option 4) and press again **Return**.

```
 1.20  EDIT SHARE VARIABLES

 Select: 1 = Edit variables
         2 = Edit groups
         3 = Edit alternatives
         4 = End
           1

 Select: Operation on variables
         1 = Add
         2 = Modify
         3 = Delete
         4 = List
         5 = Edit comments
           4

 Detailed list? y

 List by group? y

 Do you want only the descriptions? y

 Enter: Name of group(s) to be listed (max. 1000 characters)
 = *
```

Send the output to the screen and if you wish to see the next group of variables and their definition just press the key "**n**" (next page). Again if you have any question concerning the scrolling commands

just press the key "**h**" during the scrolling session and a Help menu will be displayed. To exit the definition list of share variables, press the key "**q**" and answer "**no**" to the question "**Send to print file?**"

4.2 DEFINE AND INITIALIZE SHARE VARIANTS

In this section we will show how to create a series of share models: in particular a **LOGIT**, an **S-DOGIT**, and a **LIN-IPT** variant. For additional information concerning these types of models, the user should refer to the Reference and Model Types manuals.

4.2.1 Creation of share variants

Enter **module 2.20**, select "**Create variants**" and then "**Enter interactively**". The first defined variant will be nonlinear and will use the first 120 out of 155 observations available in the database. We will name this variant "**logit**".

We must first specify which alternatives are to be included in the variant. The Canada dataset is defined over four alternatives, so we will enter all the alternatives, **omitting a constant term only for alternative number 4**.

```
Select: Creation method
        1 = Copy a variant
        2 = Enter interactively
          2

Enter: Name of variant to be created = logit

Enter: Alternative number for equation  1 = 1

Introduce constant? y

Enter: Alternative number for equation  2 = 2

Introduce constant? y

Enter: Alternative number for equation  3 = 3

Introduce constant? y

Enter: Alternative number for equation  4 = 4

Introduce constant? n

Enter: Alternative number for equation  5 =
```

TRIO quits the selection of an alternative as soon as no alternative number is entered and then presents the full page variant editor. Note that the commands associated with this editor are the same as the ones used in the level variant editor.

Enter the name of the dependent variable under the heading "DEPENDENT"; in this case the variable should be "**FOURMODE**". End the insertion mode by pressing "**Return**".

Now, move the cursor to the first utility function (besides "**1 C**") using the usual keys , "**s**", "**e**", "**d**" and "**x**" and insert the variable names in order to get the following model structure. Remember that

in order to enter a particular variable into the variant structure, you have to press the key "**i**" followed by the name of the variable (for example "**ODREG**") and then press "**Return**".

```
line   8col 10               VARIANT DEFINITION

         DEPENDENT     FAMILY TYPE
         FOURMODE      1  LOGIT

BOX COX  NO BC         NO BC         NO BC         NO BC         NO BC
ALTER.
 1 C     ODREG         LANGFR        REVE
 2 C                                              ODREG         LANGFR
 3 C
 4
```

Continue scrolling to the right with the "**d**" key and enter the variables for alternatives 2 and 3 as shown in the next table.

```
line  10col 66               VARIANT DEFINITION

         DEPENDENT     FAMILY TYPE
         FOURMODE      1  LOGIT

BOX COX  NO BC         NO BC         NO BC         NO BC         NO BC
ALTER.
 1 C
 2 C     ODREG         LANGFR        REVE
 3 C                                              ODREG         LANGFR
 4
```

Move the cursor to the right of variable "LANGFR" of alternative 3 and insert the following columns. The screen should scroll to the left and when the last variable has been entered you should obtain the following table:

```
line   8col 10               VARIANT DEFINITION

         DEPENDENT     FAMILY TYPE
         FOURMODE      1  LOGIT

BOX COX  NO BC         NO BC         NO BC         NO BC         NO BC
ALTER.
 1 C                                 RETNLSC1      RFANLSC1      FREQU1
 2 C                                 RETNLSC2      RFANLSC2      FREQ2
 3 C     REVE                        RETNLSC3      RFANLSC3      FREQ3
 4                     NUITA
```

When the definition of the variant is completed, **save and quit** the new variant using the key "**k**".

4.2.2 Initialization of share variants

As for probability variants, you must initialize the share variants in order to be able to estimate them later (answer "**yes**" to the question "**Initialize the variant?**").

```
Initialize variant? y
Enter: First estimation observation (    1) =

Enter: Last estimation observation (  155) = 120

Select: 1 = No constraint on the elements of SIGMA
            All the elements are estimated.
        2 = Constraint: all the off-diagonal elements of SIGMA are
            set to zero. Only the diagonal elements are estimated.
        3 = Constraint: all the off-diagonal elements of SIGMA are
            set to zero. Constraint: all the diagonal elements of
            SIGMA are equal and they are estimated.
          1

Do you want to modify initial values of constants? n

Modify initial values of parameters (beta,lambda)? n
Validating data ...
Validating values ...
Validating model ...
```

Initial values may be provided to start the maximization process elsewhere than at the default values of the parameters. Since no previous values of the parameters are available, **answer "no"** to the questions "Do you want to modify initial values of constants?" and "Do you want to modify initial values of parameters (beta,lambda)?".

Complete the initialization, by setting the "Maximum number of iterations in likelihood function maximisation algorithm" to **300**, and by using the default values for the "Tolerance factor in likelihood function maximisation algorithm" and "Weight for Hessian matrix". Finally **scale** the data and **save** the initialization of the variant.

```
Enter: Maximum number of iterations in likelihood
       function maximization algorithm ( 100) = 300

Enter: Tolerance factor in likelihood function
       maximization algorithm (.10E-04) =

Enter: Weight for the Hessian (.10E+01) =

Scaling?  y

Save initialization? y

Initializing variant ...

Conditional least-squares will be computed
Variant "logit:0" has been initialized.

Save new variant? y
```

At the end of the initialization exit the module by selecting "**No futher edition**" and then "**End**".

4.3 ESTIMATE VARIANTS

Once the share variant "**logit**" has been successfuly initialized you can proceed with its estimation.

4.3.1 Estimation waiting list

For the estimation of share variants we use the same process ("**Add, Delete, List**") as for probability and level variants. Enter **module 3.20** and select the option "**Add**". Then enter the name of the only initialized variant (**logit**) in the estimation list.

```
3.20  ESTIMATE SHARE VARIANTS

Select: Operation on variants to be estimated
        1 = Add
        2 = List
        3 = Delete
        4 = Estimate
        5 = End
          1

Enter: Name[:version] of variant to be estimated = logit

Enter: Name[:version] of variant to be estimated =
```

Let us make a list of the variants present in the estimation list.

```
 3.20  ESTIMATE SHARE VARIANTS

 Select: Operation on variants to be estimated
         1 = Add
         2 = List
         3 = Delete
         4 = Estimate
         5 = End
           2
```

The list of variants to be estimated should look like this:

```
              LIST OF VARIANTS TO BE ESTIMATED
              ________________________________



 Present time = 14:28:58


 No     Variant      Version   Status   Start date   Start time   End time
 __   ___________   _______   ______   __________   __________   ________


 1   logit              0       I
WAIT. Press any key to continue.
```

4.3.2 Run estimation procedure

To estimate the variant select "**Estimate**" from the "**Operation on variants to be estimated**" menu and follow the instructions.

```
 3.20  ESTIMATE SHARE VARIANTS

 Select: Operation on variants to be estimated
         1 = Add
         2 = List
         3 = Delete
         4 = Estimate
         5 = End
           4
```

TRIO will ask you again what kind of preliminary report you wish to have and if the estimation procedure is to be run in background or not. For expositional purposes we will answer "**Short report**", send the report to the "**Terminal**", and "**no**" to the background estimation.

```
Select: Type of report
        1 = No    report
        2 = Short report
        3 = Long  report
          2

Do you want to run your estimation(s) in "background" ? n

Select: List device
        1 = Terminal
        2 = Print file
          1
```

A short report concerning the maximization steps of the objective function will be displayed on the screen. At the end of the maximisation process answer "**no**" to the question "Send to print file?". Finally exit the estimation module by selecting the "**End**" option.

4.3.3 Using an estimated variant as a starting point

We will now show how to use an estimated variant solution (the convergent solution) as a starting point for other more general variants. In this way, the convergence time of the new variants will be greatly reduced. As an example, we will create two new variants; a "S-DOGIT" type variant and a "BT-IPT" type variant. Let us start with the "S-DOGIT" variant.

4.3.3.1 The S-Dogit variant

Again enter **module 2.20**, select "**Create variants**" and then "**Copy a variant**". Enter the name of the variant to be created (we will name this variant "**sdogit**").

```
Enter: Name of variant to be created = sdogit
```

Enter the name of the variant to be copied ("**logit:0**").

```
Enter: Name[:version] of variant to be copied = logit:0
```

And the following message will appear.

```
Variant "sdogit:0" now exists in database.
```

Please remember that variants created by the copying mechanism only differ from the original ones by their name (they keep the same status of the original variants; C, I or E).

Now you must modify variant "sdogit" in the following way:

```
 2.20  EDIT SHARE VARIANTS

 Select: 1 = Create     variants
         2 = Modify     variants
         3 = Initialize variants
         4 = Delete     variants
         5 = List       variants
         6 = See variant contents
         7 = Edit comments
         8 = End
           2

 Enter: Name[:version] of variant to be modified = sdogit

 Using estimated results as initial values for parameters? y

 Modify definition? y

 Modify alternatives or constants? n
```

The initial variant definition page is the same as the one shown during the creation of variant "logit:0":

```
line   4col 10              VARIANT DEFINITION

         DEPENDENT     FAMILY TYPE
         FOURMODE      1  LOGIT

BOX COX  NO BC         NO BC         NO BC         NO BC         NO BC
ALTER.
 1 C     ODREG         LANGFR        REVE
 2 C                                               ODREG         LANGFR
 3 C
 4
```

Place the cursor in the "**FAMILY TYPE**" field under the heading "**1 LOGIT**" using the key "**d**" and press "**i**" followed by the number "**2**". The following modification will take place:

```
line   4col 24              VARIANT DEFINITION

         DEPENDENT     FAMILY TYPE
         FOURMODE      2  STANDARD DOGIT

BOX COX  NO BC         NO BC         NO BC         NO BC         NO BC
ALTER.
 1 C     ODREG         LANGFR        REVE
 2 C                                               ODREG         LANGFR
 3 C
 4
```

Quit the share variant editor saving the new variant by pressing the key "**k**" and proceed with the initialization of the variant.

```
Initialize variant? y

Enter: First estimation observation (    1) =

Enter: Last estimation observation (  155) = 120

Select: 1 = No constraint on the elements of SIGMA
            All the elements are estimated.
        2 = Constraint: all the off-diagonal elements of SIGMA are
            set to zero. Only the diagonal elements are estimated.
        3 = Constraint: all the off-diagonal elements of SIGMA are
            set to zero. Constraint: all the diagonal elements of
            SIGMA are equal and they are estimated.
          1

Do you want to modify initial values of constants? n

Modify initial values of parameters (beta,lambda)? n

Modify initial values of extra parameters (mu, phi or theta)? y
```

The following menu will appear (For more information concerning the meaning of the “Theta values”, the “Group Structure” and the “Status” you should refer to the Reference Manual):

```
            INITIALIZATION OF THETA PARAMETERS IN  SHARE VARIANT

            THETA NO       INITIAL          GROUP          STATUS
                           THETA VALUE      STRUCTURE

                1           0.00000E+00         0            E
                2           0.00000E+00         0            E
                3           0.00000E+00         0            E
                4           0.00000E+00         0            E
```

For the tutorial purpose, accept these default values by saving them (press the key “**k**”) and then, enter the following answers:

```
Validating data ...
Validating values ...
Validating model ...

Enter: Maximum number of iterations in likelihood
       function maximization algorithm ( 300) =

Enter: Tolerance factor in likelihood function
       maximization algorithm (.10E-04) =

Enter: Weight for the Hessian (.10E+01) =

Scaling?  y

Save initialization? y

Initializing variant ...

Conditional least-squares will be computed
```

At the end of the initialization exit the module by selecting "**No further edition**".

4.3.3.2 The BT-IPT variant

Let us now create our last variant. Again select "**Create variants**" (from **module 2.20**) and then "**Copy a variant**". This time, name the variant to be created "**btipt**". Then enter the name of the variant to be copied ("**logit:0**")

```
Enter: Name of variant to be created = btipt

Enter: Name[:version] of variant to be copied = logit:0

Variant "btipt:0" now exists in database.
```

Proceed with the modification of the variant just created in the same way as for variant "sdogit"

```
 2.20  EDIT SHARE VARIANTS

 Select: 1 = Create     variants
         2 = Modify     variants
         3 = Initialize variants
         4 = Delete     variants
         5 = List       variants
         6 = See variant contents
         7 = Edit comments
         8 = End
           2

 Enter: Name[:version] of variant to be modified = btipt

 Using estimated results as initial values for parameters? y

 Modify definition? y

 Modify alternatives or constants? n
```

Again the initial variant definition page is identical to the one shown during the creation of variant "logit:0":

```
line   4col 24               VARIANT DEFINITION

         DEPENDENT     FAMILY TYPE
         FOURMODE      1 LOGIT

BOX COX  NO BC         NO BC         NO BC         NO BC         NO BC
ALTER.
 1 C     ODREG         LANGFR        REVE
 2 C                                                ODREG         LANGFR
 3 C
 4
```

Place the cursor under the heading "FAMILY TYPE" using the key "**d**" and press "**i**" followed by the number "**5**". The following modification will take place;

```
line   4col 24               VARIANT DEFINITION

         DEPENDENT     FAMILY TYPE
         FOURMODE      5 BT-IPT-LOGIT

BOX COX  NO BC         NO BC         NO BC         NO BC         NO BC
ALTER.
 1 C     ODREG         LANGFR        REVE
 2 C                                                ODREG         LANGFR
 3 C
 4
```

Save the new variant and initialize it using the same maximisation parameters already used during the initialization of variant "**sdogit**".

```
Initialize variant? y

Enter: First estimation observation (    1) =

Enter: Last estimation observation (  155) = 120

Select: 1 = No constraint on the elements of SIGMA
            All the elements are estimated.
        2 = Constraint: all the off-diagonal elements of SIGMA are
            set to zero. Only the diagonal elements are estimated.
        3 = Constraint: all the off-diagonal elements of SIGMA are
            set to zero. Constraint: all the diagonal elements of
            SIGMA are equal and they are estimated.
          1

Do you want to modify initial values of constants? n

Modify initial values of parameters (beta,lambda)? n

Modify initial values of extra parameters (mu, phi or theta)? y
```

The following menus will appear (For more information concerning the meaning of the "Phi and Mu values", the "Group Structure" and the "Status" you should refer to the Reference Manual):

```
            INITIALIZATION OF PHI PARAMETERS IN SHARE VARIANT

              PHI NO       INITIAL          GROUP          STATUS
                           PHI VALUE        STRUCTURE

                1          0.00000E+00        0              E
                2          0.00000E+00        0              E
                3          0.00000E+00        0              E
                4          0.00000E+00        0              E
```

For tutorial purposes, accept these default values for the "**Phi**" parameters by saving them (press the key "**k**").

Another menu, this time concerning the "**Mu**" parameters, will be displayed. Please **do the following modifications**. Place the cursor under the heading "GROUP STRUCTURE and assign the group value "**1**" to **all the "Mu" parameters** by pressing the key "**i**" and inserting the "NEW VALUE = **"1"** followed by "**Return**" (**remember to do this for all the Mu parameters of this variant**). At the end of the modifications you should obtain the following table.

```
          INITIALIZATION OF MU PARAMETERS IN SHARE VARIANT

             MU NO        INITIAL          GROUP          STATUS
                          MU VALUE         STRUCTURE

               1           0.00000E+00         1            E
               2           0.00000E+00         1            E
               3           0.00000E+00         1            E
               4           0.00000E+00         1            E
```

Press the key "**k**" and continue with the initialization of the variant.

```
Validating data ...
Validating values ...
Validating model ...

Enter: Maximum number of iterations in likelihood
       function maximization algorithm ( 300) = 400

Enter: Tolerance factor in likelihood function
       maximization algorithm (.10E-04) =

Enter: Weight for the Hessian (.10E+01) =

Scaling?  y

Save initialization? y

Initializing variant ...

Conditional least-squares will be computed
```

At the end of the initialization exit the module by selecting "**No further edition**" and then the "**End**" option

4.3.4 Estimating the new variants

Enter **module 3.20** and select the option "**Add**". Then enter the name of the new variant initialized, "**sdogit:1**" and "**btipt:1**" in the estimation list. Now, in order to estimate these variants, simply select "**Estimate**" from the "**Operation on variants to be estimated**" menu and enter the following answers.

```
3.20  ESTIMATE SHARE VARIANTS

Select: Operation on variants to be estimated
        1 = Add
        2 = List
        3 = Delete
        4 = Estimate
        5 = End
          4
```

```
Select: Type of report
        1 = No    report
        2 = Short report
        3 = Long  report
          2

Do you want to run your estimation(s) in "background" ? n

Select: List device
        1 = Terminal
        2 = Print file
          1
```

The maximization steps of the objective functions will be displayed on the screen. At the end of the maximisation process answer "**no**" to the question "Send to print file?"

4.4 ANALYZING VARIANT RESULTS

Now that variants "logit:0", "sdogit:1" and "btipt:1" have been estimated we can proceed with the analysis of results.

4.4.1 Variant comparisons with TABLEX

The share TABLEX follows the same general structure as that of the level and probability TABLEX. The output is modified to accomodate both the multiple alternatives and the model families of the "Share Class" (the results are again listed by alternative). For each variable appearing in the utility functions, eight results are available (some of them are vectors of values instead of a single value). Since the available results are different for this type of model class, the appearance of the table is modified. However the same column to column display is always used.

4.4.1.1 Reading a set of variants from an ascii file

We will now introduce in our database a group of three others variants included in the variant input batch file "**d221.in**" of the Tutorial database. These variants will be created, initialized and estimated by TRIO during the batch reading process.

In particular a **Box-Cox Logit** model, a **Generalized Dogit** model and a **Lin-Ipt-Logit** model will be estimated in batch mode. For additional information concerning all these models you should refer to the Reference and Model Type manuals.

Enter **module 2.21** and select "**Read variants from ascii file**". Then press "**Return**" to accept the default file named **d221.in**. **To accelerate the process, do not echo the batch reading**. After the reading process is terminated, press "**q**" twice and you will exit this module.

4.4.1.2 The TABLEX construction

Prior to creating the share TABLEX, we normally need to define the groups of variables and labels corresponding to each of the alternatives in order to facilitate the reading of the TABLEX output. Because we have already done this previously using TRIO's batch capability, we can proceed immediately with the creation of our first share TABLEX

To do so enter **module 4.25** and select the **TABLEX** option.

```
Select: Printing format
        1 = normal
        2 = compressed
          2

Select: Page  width
        1 =  80 columns
        2 = 128 columns
        3 = 160 columns
          2

Select: List device
        1 = Terminal
        2 = Print file
          1

Do you want the default number of lines per page (  70 ) ? y

Enter: Table title (SHARE VARIANT COMPARISON) =
```

Then a list of the available statistics for this type of TABLEX will be displayed. Please, **select options 3, 2 and 8**.

```
Enter: Data to be listed

        1  =  Regression coefficients
        2  =  Gradient

        3  =  Elasticities  at the sample mean of X (direct/cross)
        4  =  Change in P(i) relative to percent change in X(j,k)
              at the sample mean of X (direct/cross)
        5  =  Derivative of P(i) with respect to X(j,k) in
              original units at the sample mean of X (direct/cross)
        6  =  Marginal rate substitution at the sample mean of X

        7  =  Unconditional T-statistic
        8  =  Conditional T-statistic

Enter with a space between the items : 3 2 8
```

In order to reduce the TABLEX creation time **do not select cross results**.

```
For ELASTICITIES AT MEAN
Do you want the cross data list? n

RESULT TO BE LISTED :  3   2   8
```

Then enter the variant names and the alternatives to be displayed as follows:

```
VARIANTS :
----------
Enter: Name of   1 variant[:version] : logit:0
Enter: Name of   2 variant[:version] : bclogit:1
Enter: Name of   3 variant[:version] : sdogit:1
Enter: Name of   4 variant[:version] : gdogit:1
Enter: Name of   5 variant[:version] : linipt:1
Enter: Name of   6 variant[:version] : btipt:1
Enter: Name of   7 variant[:version] :

Alternatives to be considered :  1  2  3  4

Do you want all of these alternatives ? y

GROUPS :
--------
Do you want the default groups? y

Do you want only variables that appear in variant? y

Do you want a detailed listing of the Box-Cox transformations? n
```

The default groups defined earlier will be used, and only the variables that appear in at least one alternative will be displayed. After the initialization a trace of the construction is displayed.

```
TABLEX initialization
Proceeding
  Part I  : Alternative  1
                Variant  1
                Variant  2
                Variant  3
                Variant  4
                Variant  5
                Variant  6
            Alternative  2
                Variant  1
                Variant  2
                Variant  3
                Variant  4
                Variant  5
                Variant  6
            Alternative  3
                Variant  1
                Variant  2
                Variant  3
                Variant  4
                Variant  5
                Variant  6
            Alternative  4
                Variant  1
                Variant  2
                Variant  3
                Variant  4
                Variant  5
                Variant  6
  Part II : Parameters
  Part III: General Statistics

The TABLEX is finished.
```

Some tables showing partial results are given below.

```
TRIO module: 4.25      Page:   1                    Date: 93/11/08                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------

                              SHARE VARIANT COMPARISON

=============================================================================================
I.  ELASTICITIES AT MEAN        TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
    GRADIENT                 VARIANT = logit   bclogit   sdogit   gdogit   linipt   btipt
    (COND.   T-STATISTIC)    VERSION =    0        1        1        1        1        1
                            DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
=============================================================================================
ALTERNATIVE   1 : AIR
====================
----------------
NETWORK VARIABLES
----------------

  PLANE FARE / CAR FARE  RFANLSC1        -.670    -.716   -2.932   -3.291   -2.436    -.482
                                       .39E-05  .33E-03  .24E-03  .18E-03  .81E-03  .12E-02
                         (1)           (-6.90)  (-2.55)  (-6.36)  (-4.91)  (-5.57)  (-2.51)
                                          (GE) L  2(GE)     (GE)     (GE)     (GE)     (GE)

  TIME BY CAR /          RETNLSC1        1.782    1.410     .894    1.146    1.434    1.736
  TIME BY PLANE                       -.22E-05  .67E-03 -.21E-02  .41E-03  .39E-03  .61E-02
                         (1)             (7.93)   (7.05)   (3.22)   (2.80)   (2.25)   (5.65)
                                          (GE) L  2(GE)     (GE)     (GE)     (GE)     (GE)

  PLANE                  FREQU1           .247     .218     .307     .485     .257     .061
  UTILITARY FREQUENCY                   .63E-05 -.84E-02  .22E-02 -.52E-02  .45E-02  .10E+00
                         (1)             (4.95)   (5.39)   (4.90)   (3.95)   (5.01)   (3.67)
                                          (GE) L  3(GE)     (GE)     (GE)     (GE)     (GE)

-----------------------
SOCIOECONOMIC VARIABLES
-----------------------

  LANGUAGE               LANGFR           .018     .227     .401     .829     .682     .117
  SIMILARITY O-D                        .28E-04  .30E-02 -.28E-01  .10E-01  .11E-01  .35E-01
                         (1)             (.13)   (1.65)   (1.46)   (2.02)   (1.72)    (.55)
                                          (SP) L  1(SP)     (SP)     (SP)     (SP)     (SP)

  O-D WITHIN             ODREG           -.144    -.125    -.080    -.335    -.331    -.043
  THE SAME PROVINCE      =====          .69E-06  .18E-03  .60E-04  .85E-04  .13E-03  .30E-04
                         (1)            (-.57)   (-.41)   (-.26)   (-.28)   (-.97)    (.04)
                                          (SP)     (SP)     (SP)     (SP)     (SP)     (SP)

  INCOME                 REVE            1.310    1.432    3.535    4.482    4.805    1.949
                                       .10E-02  .11E-01 -.17E+01  .69E+00  .76E+00  .35E+01
                         (1)             (.89)    (.98)   (1.91)   (1.76)   (2.38)   (1.56)
                                          (SP) L  1(SP)     (SP)     (SP)     (SP)     (SP)

---------------------------------------------------------------------------------------------

  REGRESSION CONSTANT    CONSTANT          -        -        -        -        -        -
                                       .25E-06  .43E-03 -.34E-03  .16E-03  .18E-03  .94E-03
                         (1)           (-2.66)  (-1.53)  (-2.38)  (-2.21)  (-2.87)  (-3.68)
```

```
TRIO module: 4.25      Page:   2                    Date: 93/11/08                                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------------

================================================================================
I.  ELASTICITIES AT MEAN        TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
    GRADIENT                 VARIANT = logit   bclogit   sdogit   gdogit   linipt   btipt
    (COND.   T-STATISTIC)    VERSION =    0        1        1        1        1        1
                            DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
================================================================================
ALTERNATIVE   2 : RAIL
======================
-----------------
NETWORK VARIABLES
-----------------

                        RFANLSC2        -.281    -.949   -1.127   -1.220   -1.059    -.903
                                       .39E-05  .33E-03  .24E-03  .18E-03  .81E-03  .12E-02
                        (2)           (-6.90)  (-2.55)  (-6.36)  (-4.91)  (-5.57)  (-2.51)
                                         (GE) L  2(GE)     (GE)     (GE)     (GE)     (GE)

                        RETNLSC2         .415    1.878     .191     .236     .346    1.805
                                      -.22E-05  .67E-03 -.21E-02  .41E-03  .39E-03  .61E-02
                        (2)             (7.93)   (7.05)   (3.22)   (2.80)   (2.25)   (5.65)
                                         (GE) L  2(GE)     (GE)     (GE)     (GE)     (GE)

                        FREQ2            .183     .169     .208     .317     .197     .203
                                      .63E-05 -.84E-02  .22E-02 -.52E-02  .45E-02  .10E+00
                        (2)             (4.95)   (5.39)   (4.90)   (3.95)   (5.01)   (3.67)
                                         (GE) L  3(GE)     (GE)     (GE)     (GE)     (GE)

------------------------
SOCIOECONOMIC VARIABLES
------------------------

                        LANGFR          -.077    -.048    -.141    -.272    -.172    -.127
                                      .13E-04 -.32E-02 -.57E-02 -.46E-02 -.10E-01 -.27E+00
                        (2)             (-.16)    (.34)   (-.17)   (-.52)    (.05)   (-.05)
                                         (SP) L  1(SP)     (SP)     (SP)     (SP)     (SP)

                        ODREG           -.338     .287    -.066     .117    -.159     .513
                        =====        -.15E-06 -.20E-03  .22E-04 -.73E-05 -.53E-04 -.29E-02
                        (2)           (-1.41)   (1.05)   (-.27)    (.55)   (-.94)   (2.13)
                                         (SP)     (SP)     (SP)     (SP)     (SP)     (SP)

                        REVE            3.369    1.405    2.974     .479    2.998    -.102
                                      .15E-02 -.11E-01 -.85E+00 -.38E+00 -.50E+00 -.28E+02
                        (2)             (1.95)   (1.07)   (1.86)    (.18)   (2.24)    (.59)
                                         (SP) L  1(SP)     (SP)     (SP)     (SP)     (SP)

-------------------------------------------------------------------------------------------

  REGRESSION CONSTANT   CONSTANT           -        -        -        -        -        -
                                      .37E-06 -.47E-03 -.17E-03 -.91E-04 -.99E-04 -.70E-02
                        (2)           (-3.66)  (-1.51)  (-3.04)   (-.96)  (-3.65)  (-4.53)
```

```
TRIO module: 4.25      Page:   3                    Date: 93/11/08                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------------

================================================================================================
I.  ELASTICITIES AT MEAN          TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
    GRADIENT                   VARIANT = logit   bclogit   sdogit   gdogit   linipt   btipt
    (COND.   T-STATISTIC)      VERSION =    0        1        1        1        1        1
                              DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
================================================================================================
ALTERNATIVE   3 : BUS
====================
-----------------
NETWORK VARIABLES
-----------------

                          RFANLSC3        -.257    -.968   -1.034    -.969    -.959   -1.128
                                         .39E-05  .33E-03  .24E-03  .18E-03  .81E-03  .12E-02
                          (3)            (-6.90)  (-2.55)  (-6.36)  (-4.91)  (-5.57)  (-2.51)
                                            (GE) L  2(GE)     (GE)     (GE)     (GE)     (GE)

                          RETNLSC3         .397    1.915     .183     .196     .328    2.361
                                        -.22E-05  .67E-03 -.21E-02  .41E-03  .39E-03  .61E-02
                          (3)             (7.93)   (7.05)   (3.22)   (2.80)   (2.25)   (5.65)
                                            (GE) L  2(GE)     (GE)     (GE)     (GE)     (GE)

                          FREQ3            .433     .430     .494     .653     .462     .657
                                         .63E-05 -.84E-02  .22E-02 -.52E-02  .45E-02  .10E+00
                          (3)             (4.95)   (5.39)   (4.90)   (3.95)   (5.01)   (3.67)
                                            (GE) L  3(GE)     (GE)     (GE)     (GE)     (GE)

------------------------
SOCIOECONOMIC VARIABLES
------------------------

                          LANGFR           .620     .496     .578    -.761     .532     .730
                                        -.13E-04 -.12E-02  .58E-01 -.66E-02  .63E-02  .54E+00
                          (3)             (2.46)   (3.81)   (2.52)  (-1.15)   (2.77)   (2.90)
                                            (SP) L  1(SP)     (SP)     (SP)     (SP)     (SP)

                          ODREG            .056     .530     .144    1.705     .090     .808
                          =====         .16E-06 -.91E-04  .19E-04 -.73E-04  .69E-04  .50E-03
                          (3)             (-.04)   (2.36)    (.45)   (2.17)   (-.02)   (3.67)
                                            (SP)     (SP)     (SP)     (SP)     (SP)     (SP)

                          REVE             .403   -1.370    -.172   -4.532    -.383   -2.189
                                        -.21E-02 -.43E-02  .38E+01 -.38E+00  .47E+00  .41E+02
                          (3)              (.61)    (-.46)    (.37)  (-1.68)    (.62)   (-.59)
                                            (SP) L  1(SP)     (SP)     (SP)     (SP)     (SP)

  REGRESSION CONSTANT     CONSTANT          -        -        -        -        -        -
                                        -.48E-06 -.17E-03  .83E-03 -.94E-04  .10E-03  .98E-02
                          (3)            (-3.58)   (-.45)  (-2.71)    (.43)  (-3.15)  (-6.09)
```

```
TRIO module: 4.25      Page:   4                    Date: 93/11/04                                    User: ....
 Project: Tutorial Demonstration Database
 -------------------------------------------------------------------------------------------------------------------------------

 ================================================================================================
 I.  ELASTICITIES AT MEAN          TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
     GRADIENT                   VARIANT = logit   bclogit   sdogit   gdogit   linipt   btipt
     (COND.   T-STATISTIC)      VERSION =    0        1        1        1        1        1
                               DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
 ================================================================================================
 ALTERNATIVE   4 : CAR
 ====================
 ------------------------
 SOCIOECONOMIC VARIABLES
 ------------------------

                           NUITA           -.404    -.432    -.315    -.235    -.282    -.904
                           -----          .26E-05  .21E-03 -.68E-03 -.54E-04 -.86E-04 -.88E-03
                           (4)            (-7.43)  (-6.60)  (-4.46)  (-1.25)  (-4.48)  (-4.68)
                                             (SP)     (SP)     (SP)     (SP)     (SP)     (SP)
```

For each variable, beside the description and the name, the results are displayed on different lines for all the variants. The number of the current alternative is displayed under the variable name. The

status of the variable is indicated on the last results line beside the Box-Cox transformation parameter number. The status is indicated by “GE” for generic variables, and by “SP” for specific variables. The results are compared columnwise within each alternative.

You can obtain two types of results other than values. When a dash (-) appears it signifies that the statistic requested was not calculated by TRIO, and when a series of asterisks (*) appears it implies that the statistic requested was calculated but the value was too large to be printed. If a constant term has been included for the current alternative, the available results will be displayed. At the end of the table, general statistics are available. Note that “-” indicates that a result is not applicable. For more information concerning these statistics, please refer to the Reference manual and the Model Type manual.

A table summarizing the results concerning the Box-Cox and Extra parameters (“Thetas, Mus and Phis”) is given below. Again for a detailed discussion concerning these parameters, you should refer to the Reference and Model Types manuals.

```
================================================================================================
II. PARAMETERS                    TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
    UNCOND.[T-STATISTIC=0]    VARIANT = logit   bclogit   sdogit   gdogit   linipt   btipt
    UNCOND.[T-STATISTIC=1]    VERSION =    0         1        1        1        1        1
                             DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
================================================================================================
-----------------------
BOX-COX TRANSFORMATIONS
-----------------------

  LAMBDA(X) - GROUP  1   LAM  1                    .224
                                                 [.42]
                                               [-1.45]

  LAMBDA(X) - GROUP  2   LAM  2                    -.008
                                                 [-.09]
                                               [-10.67]

  LAMBDA(X) - GROUP  3   LAM  3                   1.088
                                                [2.97]
                                                 [.24]

----------------
EXTRA PARAMETERS
----------------

  THETA   1   AIR                                        .12E-02
                                                          [2.77]

  THETA   2   RAIL                                       .25E-11
                                                            [.00]

  THETA   3   BUS                                        .41E-10
                                                            [.00]

  THETA   4   CAR                                        .97E-02
                                                            [.60]

  PHI     1   AIR                                                                .38E+01 -.13E+01
                                                                                  [1.31]  [-1.76]
                                                                                   [.97]  [-3.10]
                                                                                  GRP  1
```

```
TRIO module: 4.25      Page:   5                    Date: 93/11/08                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------------

================================================================================================
II. PARAMETERS                    TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
    UNCOND.[T-STATISTIC=0]    VARIANT = logit    bclogit   sdogit   gdogit   linipt   btipt
    UNCOND.[T-STATISTIC=1]    VERSION =     0        1        1        1        1        1
                            DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
================================================================================================
  PHI     2   RAIL                                                          .38E+01 -.25E+01
                                                                             [1.31]  [-2.19]
                                                                              [.97]  [-3.05]
                                                                             GRP  1

  PHI     3   BUS                                                           .38E+01 -.30E+01
                                                                             [1.31]  [-2.50]
                                                                              [.97]  [-3.33]
                                                                             GRP  1

  PHI     4   CAR                                                           .38E+01 -.28E+00
                                                                             [1.31]   [-.61]
                                                                              [.97]  [-2.78]
                                                                             GRP  1

  MU      1   AIR                                                           .10E+01  .15E+00
                                                                            2710.90]   [1.48]
                                                                            [-1.74]  [-8.34]
                                                                             GRP  1   GRP  1

  MU      2   RAIL                                                          .10E+01  .15E+00
                                                                            2710.90]   [1.48]
                                                                            [-1.74]  [-8.34]
                                                                             GRP  1   GRP  1

  MU      3   BUS                                                           .10E+01  .15E+00
                                                                            2710.90]   [1.48]
                                                                            [-1.74]  [-8.34]
                                                                             GRP  1   GRP  1

  MU      4   CAR                                                           .10E+01  .15E+00
                                                                            2710.90]   [1.48]
                                                                            [-1.74]  [-8.34]
                                                                             GRP  1   GRP  1

  THETAS  1   AIR
              1 , 1 AIR                                          .00E+00
                                                                  FIXED

              1 , 2 RAIL                                         .89E-01
                                                                 [1.84]

              1 , 3 BUS                                          .13E-01
                                                                  [.83]

              1 , 4 CAR                                          .00E+00
                                                                  [.00]

  THETAS  2   RAIL
              2 , 1 AIR                                          .22E-01
                                                                 [2.80]

              2 , 2 RAIL                                         .00E+00
                                                                  FIXED
```

```
TRIO module: 4.25      Page:   6                    Date: 93/11/08                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------------------

================================================================================
II. PARAMETERS                  TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
    UNCOND.[T-STATISTIC=0]   VARIANT = logit   bclogit   sdogit   gdogit   linipt   btipt
    UNCOND.[T-STATISTIC=1]   VERSION =     0        1        1        1        1        1
                           DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
================================================================================
              2 , 3 BUS                                          .86E-01
                                                                   [.74]

              2 , 4 CAR                                          .00E+00
                                                                   [.00]

  THETAS  3   BUS
              3 , 1 AIR                                          .14E-01
                                                                  [4.64]

              3 , 2 RAIL                                         .20E-09
                                                                   [.00]

              3 , 3 BUS                                          .00E+00
                                                                   FIXED

              3 , 4 CAR                                          .43E-12
                                                                   [.00]

  THETAS  4   CAR
              4 , 1 AIR                                          .49E-01
                                                                  [2.51]

              4 , 2 RAIL                                         .71E-10
                                                                   [.00]

              4 , 3 BUS                                          .64E+01
                                                                  [1.23]

              4 , 4 CAR                                          .00E+00
                                                                   FIXED
```

At the end of the TABLEX the user will find general information associated with each of the estimated variants such as their respective log-likelihood value. Again refer to the Reference manual for more information concerning these statistics. Note that the six variants correspond to the results found in Table 4 of Model Types, version 1.0, Ch.9.

```
TRIO module: 4.25      Page:   7                    Date: 93/11/08                                    User: ....
Project: Tutorial Demonstration Database
-------------------------------------------------------------------------------------------------------------------

================================================================================================
III.GENERAL STATISTICS          TYPE = LOGIT    LOGIT   S-DOGIT  G-DOGIT  LIN-IPTL BT-IPTL
                              VARIANT = logit   bclogit   sdogit   gdogit   linipt   btipt
                              VERSION =    0        1        1        1        1        1
                             DEP.VAR. =FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE FOURMODE
================================================================================================
LOG-LIKELIHOOD -FINAL VALUE           -527.456 -499.345 -509.271 -499.224 -508.794 -490.465

RHO-SQUARED              OVERALL          .929     .953     .946     .951     .952     .946
                       1 AIR              .944     .965     .962     .967     .967     .959
                       2 RAIL             .065     .205     .082     .092     .109     .183
                       3 BUS              .234     .342     .244     .232     .247     .326
                       4 CAR              .927     .952     .943     .949     .950     .946

MEAN SHARES  OBSERVED / ESTIMATED
                       1 AIR           .36/.35  .36/.36  .36/.37  .36/.38  .36/.37  .36/.37
                       2 RAIL          .04/.03  .04/.03  .04/.03  .04/.03  .04/.03  .04/.03
                       3 BUS           .03/.02  .03/.02  .03/.02  .03/.02  .03/.02  .03/.02
                       4 CAR           .57/.60  .57/.58  .57/.58  .57/.58  .57/.58  .57/.58

SAMPLE -NUMBER OF ALTERNATIVES              4        4        4        4        4        4
       -NUMBER OF OBSERVATIONS            120      120      120      120      120      120
       -FIRST OBSERVATION                   1        1        1        1        1        1
       -LAST  OBSERVATION                 120      120      120      120      120      120
       -AVAILABLE OBSERVATIONS IN VT      120      120      120      120      120      120
       -AVAIL OBS IN   1 AIR              120      120      120      120      120      120
       -AVAIL OBS IN   2 RAIL             120      120      120      120      120      120
       -AVAIL OBS IN   3 BUS              120      120      120      120      120      120
       -AVAIL OBS IN   4 CAR              120      120      120      120      120      120

NUMBER OF ESTIMATED PARAMETERS
       -BETAS .VARIABLES                   13       13       13       13       13       13
              .CONSTANTS                    3        3        3        3        3        3
              .ASSOCIATED DUMMIES           0        0        0        0        0        0
       -BOX-COX TRANSFORMATIONS             0        3        0        0        0        0
       -THETA                               0        0        4       12        0        0
       -PHI                                 0        0        0        0        1        4
       -MU                                  0        0        0        0        1        1
       -COVARIANCE MATRIX - RESIDUALS       6        6        6        6        6        6
================================================================================================
```

To end displaying the TABLEX press "**q**". Answer "**yes**" to the question, "Send to print file?", and name the output file so that at the end of this session you will be able to examine the contents of the report file in detail. Note that the six variants correspond to the results found in Table 4 of Model Types, Version 1.0, chapter 9.

4.4.2 Graphic analysis

Share variables may also be analyzed, as for level variables, using Horizontal, Scatter, Bar and Pie graphs. In this section, as an example we will show how to relate explanatory variables to variant result by using a Scatter type graph and using the results of variant "logit".

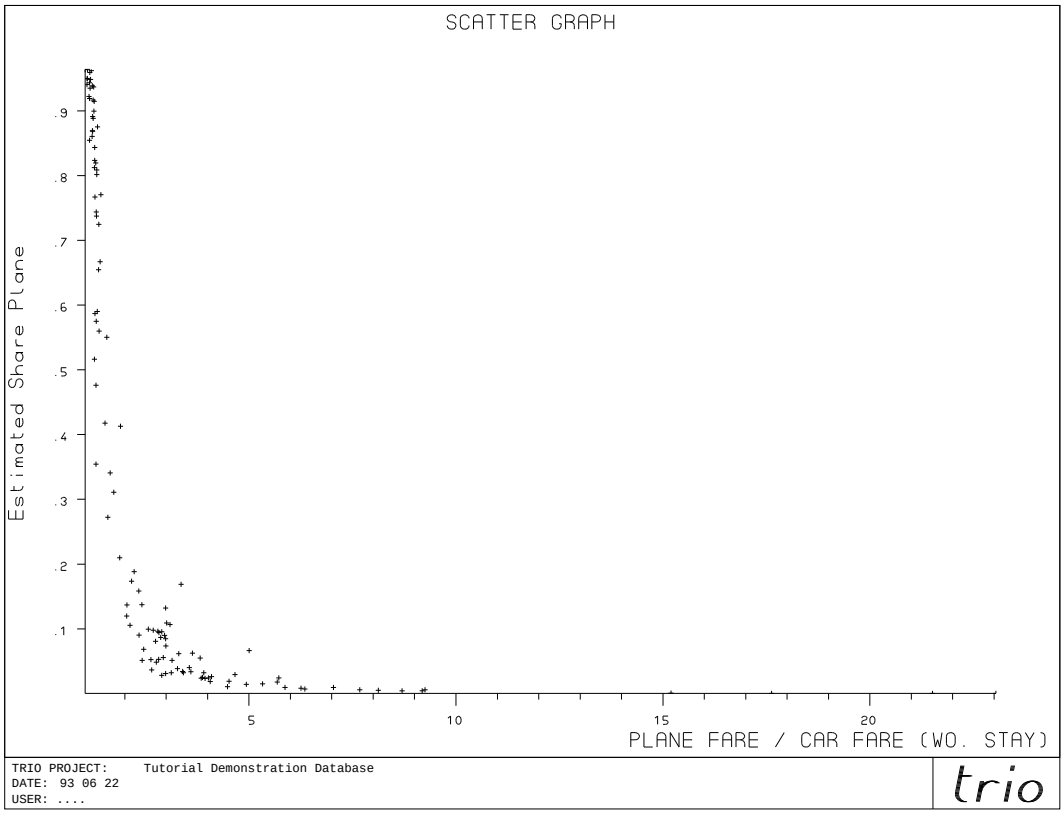
4.4.2.1 Scatter graph

Select **module 4.20**, **Make a graph** (option 1) and then follow the instructions shown in the tables.

```
 Select: Graph type
         1 = Horizontal line graph
         2 = Vertical   line graph
         3 = Scatter         graph
         4 = Bar             graph
         5 = Pie             graph
           3

 Enter: Graph title  = SCATTER GRAPH

 Enter: Character size [1 - 6: default = 2] =

 Define X axis values:

 Select: Type of values
         1 = share variables
         2 = share dependent variable
         3 = share variant results
           1

 Enter: Algebraic expression = RFANLSC1

 Enter: X axis description (PLANE FARE / CAR FARE) = PLANE FARE /
CAR FARE (WO. STAY)

 Enter: Character size [1 - 6: default = 2] =

 Define Y axis values:

 Select: Type of values
         1 = share variables
         2 = share dependent variable
         3 = share variant results
           3

 Enter: Variant name [:version] to consider = logit
```

```
Select: Values
        1 = Observed share for one observation
        2 = Observed share for one alternative
        3 = Residuals for one observation(regression errors)
        4 = Residuals for one alternative(regression errors)
        5 = Estimated share for one observation
        6 = Estimated share for one alternative
        7 = Simulated prediction for one observation
          6

Using estimated results? y

Computing SHARE ...
Enter: Alternative number = 1

Enter: Y axis description (estimated share alt) = Estimated Share
Plane

Enter: Character size [1 - 6: default = 2] =

X Y plot is defined for observation    1 to   120

Enter: First observation to consider =

Enter: Last observation to consider =

Define a constraint variable? n

Save in the temporary list of graphs? y

Plot? y
```

Answering "**yes**" to the question "**Define constraint variable?**" allows the user to filter the variables to be shown in the graphic by a third variable (explanatory, dependent or variant result). For example you may only want to see the observations associated to a revenu higher than a cetain value (which correspond to the basic idea of the conditional regression analysis). But for the purpose of this Tutorial please answer "**no**" to the "**Define a contraint variable?**" question and "**yes**" to the save and plot questions .



4.5 PERFORMING A BACKUP OF THE DATABASE

As already mentioned it is always advised to do a regular backup of the variables, groups and variants included in a database. This option is also particularly useful when variants or variables have to be transferred from one database to another.

4.5.1 Variable backup

Again we are going to perform a backup of the variables of our database, this time the share variables. To do this, exit **module 4.25**, enter in **module 1.21** and **select** the **"Write variables to ascii file"** (option **3**).

Answer "*****" to the question "Variable names to be written?" in order to save all the variables present in the database.

TRIO will then ask you "Do you want to write the descriptions and comments?". You should always answer "**yes**" to this question in order to save all the descriptions and comments entered during the TRIO session.

The next question asked is "If applicable, do you want algebraic expressions instead of values?". Again answer "**yes**" if you want to save the newly created algebraic expressions.

```
1.21  READ/WRITE SHARE VARIABLES

Select: 1 = Read variables from ascii file
        2 = Read groups and alternatives from ascii file
        3 = Write variables to ascii file
        4 = Write groups and alternatives to ascii file
        5 = End
          3

Variable names to be written (max. 1000 characters)
= *

Do you want to write the descriptions and comments? y

If applicable, do you want algebraic expressions instead of values? y
```

TRIO will then ask you for the name of the output file:

```
Enter: Name of your output file (d121.out) = Canada.in
```

When the "write" phase is over, **press Return**.

4.5.2 Groups backup

Proceed in the same way to make a backup up of the groups created in the database. **Select** the "**Write groups and alternatives to ascii file**" (option 4). Then TRIO will ask for the name of the output file:

```
Enter: Name of your output file (g121.out) = Canagrp.in
```

Answer "*****" to the question "Group names to be written" in order to save all the groups included in the database. TRIO will then ask you "Do you want to write the descriptions and the order?". You should always answer "**yes**" to this question and save all the descriptions and order number of the groups entered during the TRIO session. Then answer "**yes**" to the questions "Do you want to write the alternatives?" and "Do you want to write the description?" to save the alternatives used in the variants and their description.

When the "write" phase is over (at the "WAIT. Press any key to continue." prompt), you can **press Return** and exit **module 1.21** by typing the key "**q**" or by selecting option "**5**".

4.5.3 Variant backup

Now to do a backup of the variants present in the database enter **module 2.21**. The following dialogue of **module 2.21** shows how to backup all the probability variants (created, initialized and estimated).

```
2.21  READ/WRITE SHARE VARIANTS

Select: 1 = Read variants from ascii file
        2 = Write variants to ascii file
        3 = End
          2

Do you want to write particular variant versions? n

Enter: Name of variant(s) to be written = *
Do you want all versions ( last version if not)? y

 8 variant(s) to be written.

Do you want to write the comments? y


Enter: Name of your output file (d221.out) = share.out
```

All the variants will be written to an external ASCII file named **share.out** (output file). You can later view this file on your screen to see what it looks like.

To end the share session press "**Return**" at the "**WAIT. Press any key to continue.**" prompt and then "**3**" or "**q**". Press "**Return**" once more and the main "Variable Editor" will be displayed.

If after this short introduction to TRIO share models you have additional questions concerning some of TRIO's commands, you should consult the **Reference Manual** (or the **Model Types Manual**).

CHAPTER 5: INTRODUCTION TO TRIO GRAPHS

In this section we will show you some of the advanced graphic possibilities of the TRIO package and their utility during the analysis of variables and variants estimated in TRIO.

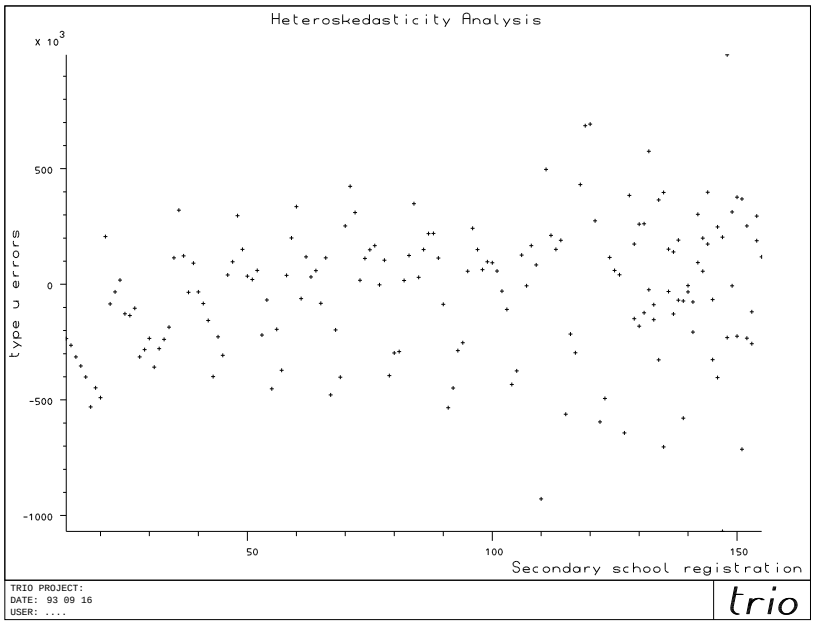
5.1 Graphic analysis

To examine some of the graphic features available in TRIO we will examine if the variable "**tend**" (school children registrations) is causing some heteroskedasticity problems in variant "**sch_bc1auhg**" (estimated in the LEVEL related chapter), and if we can effectively correct the problem by introducing a heteroskedasticity structure in the model. Let us start with the first graph. Enter **module 4.10**, select "**Make a Graph**" and then the "**Scatter graph**" option. Then enter the title "**Heteroskedasticity Analysis**".

```
Select: Graph type
        1 = Horizontal line graph
        2 = Vertical   line graph
        3 = Scatter         graph
        4 = Bar             graph
        5 = Pie             graph
          3

Enter: Graph title  = Heteroskedasticity Analysis

Enter: Character size [1 - 6: default = 2] =

Define X axis values:

Select: Type of values
        1 = level variables
        2 = level variant results
          1

Enter: Algebraic expression = tend

Enter: X axis description (SECONDARY SCHOOL REGISTRATION) = Secondary
school registration

Enter: Character size [1 - 6: default = 2] =

Define Y axis values:
```

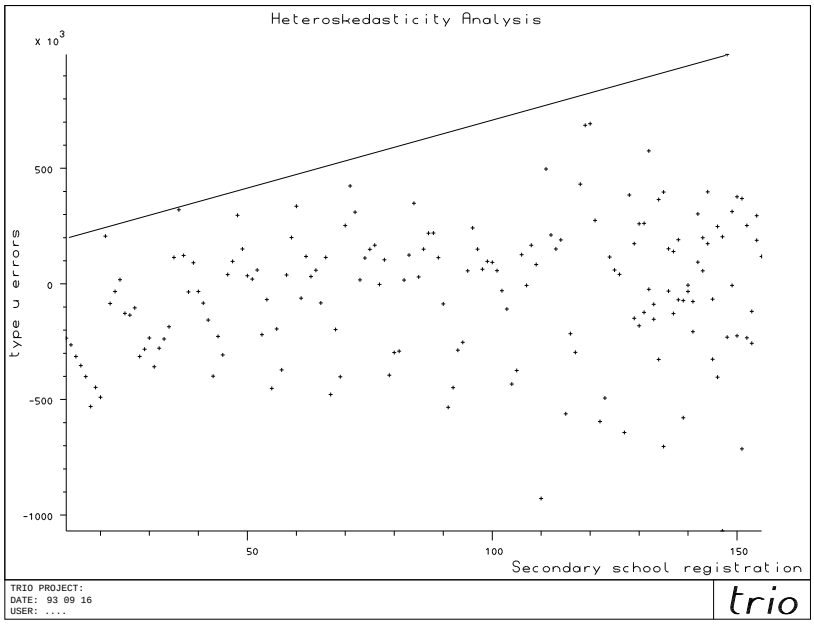
```
Select: Type of values
        1 = level variables
        2 = level variant results
          2

Enter: Variant name [:version] to consider = sch_bc1auhg

Select: Values
        1 = type u residuals (regression errors)
        2 = type u residuals in % vs a variable
        3 = type v residuals (homoskedastic errors)
        4 = type v residuals in % vs a variable
        5 = type w residuals (final model errors)
        6 = type w residuals in % vs a variable
        7 = E(Y): expected values of Y
        8 = Probability of Y not to be at the limit
        9 = E(Y**): expected values of transformed Y
       10 = Standard error of Y
       11 = Maximum likelihood forecast of E(Y)
       12 = Simulation forecast of E(Y)
       13 = Standard error of forecast
          1

Using estimated results? y

Enter: Y axis description (type u errors) =

Enter: Character size [1 - 6: default = 2] =

X Y plot is defined for period   13 to   181

Enter: First period to consider =

Enter: Last period to consider =

Define a constraint variable? n

Save in the temporary list of graphs? y

Plot? y
```

The following graph should appear on your screen.



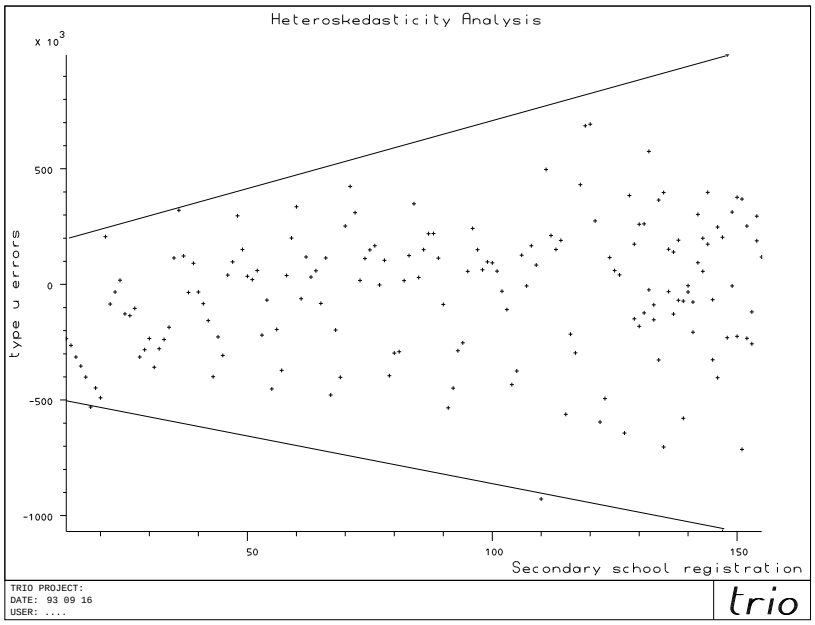
Now we will trace a line joining some of the extreme points of the graph to highlight the fact that as the number of secondary school registrations increases, the absolute value of the "**type u errors**" of the regression also increases, suggesting a problem of heteroskedasticity.

For that reason, we will add two lines to the graph. First press "**a**" in order to add an object and press the key "**p**" for "polygon", the object to be drawn. TRIO will then ask you to select the first point of the polygon. Place the cursor on the Y axis around the value 200 and press the "**space bar**" to accept the first entry. To select the last point of the polygon (the end of the first line) place the cursor on the highest point of the graph and then press the key "**a**" to accept this entry as the last point of the object (pressing the key "**a**" will also end the edition of the polygon). As a first result, you should get the following graph.

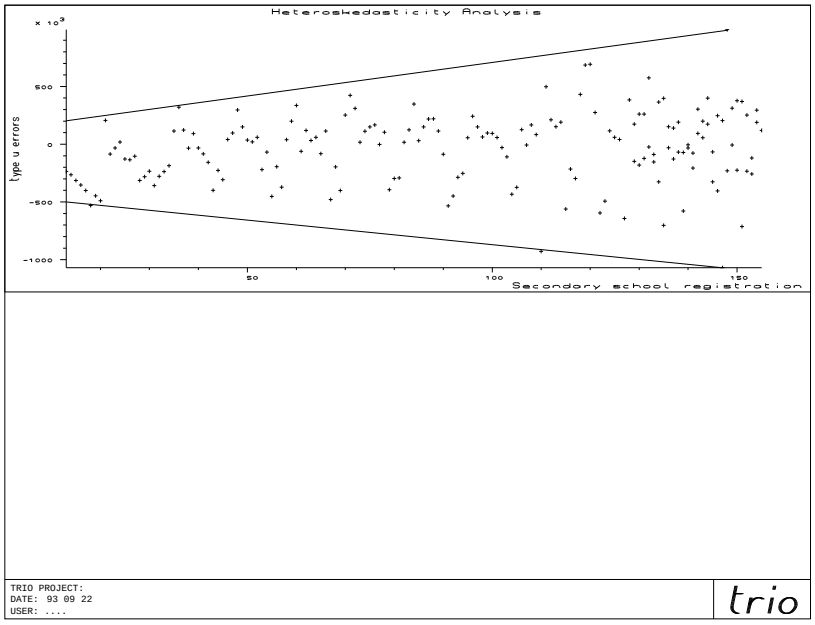


Please note that when you select a point you must always be positioned between the Y axis and the X axis. This means that you cannot select a point situated outside of the shown space.

Now let us draw the second line of the graph. Go through the same steps as for the first line but now select as a first point the value -500 on the Y axis and as the second point the lowest point (situated around the value 150 on the X axis) of the graph. Now you should get the following graph.



Now we will position the graph on the top part of the screen for presentation purposes. To do this press the key "**v**" for view and then the key "**u**" for up. For more details concerning these options please refer to the **Reference Manual**. The result of this sequence of commands should be the following graph.



Remember that if you want to save the graph in a plot file you must press the key "**k**" before pressing the key "**q**". We do not want to save the graph yet, so only press "**q**" and we will temporarily exit the graph. The user should note that as long as you do not exit module 4.10, the graph will not be lost even if you start drawing a new graph: it will be stored into the live memory of the computer – 10 successive graphs can be stored in this way and then recalled to be superimposed according to a user defined sequence.

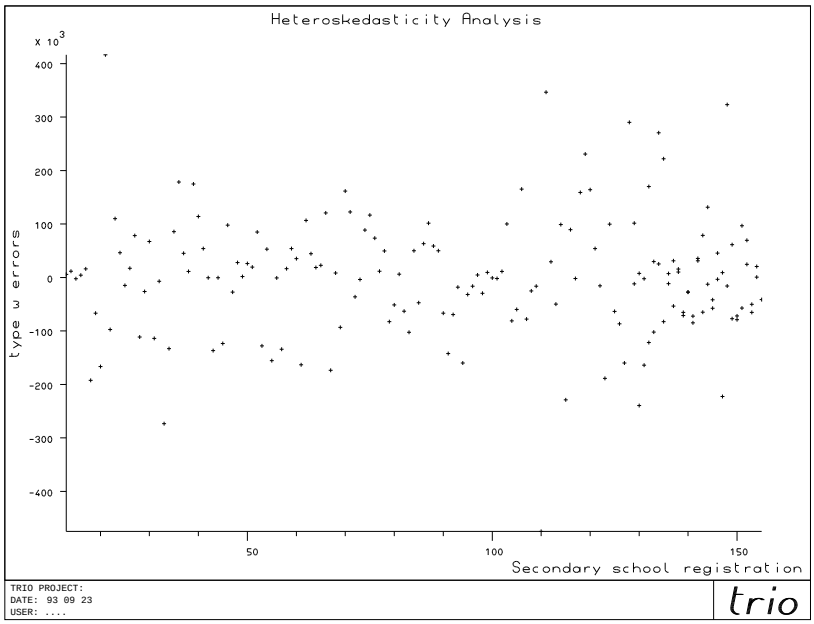
In our second graph, we will show the effect on the residuals of taking into account the heteroskedastic and autoregressive structures of variant "**sch_bc1auhg**". For this reason we will go through the same steps as for the previous set of graphs, but this time we will select the "**type w errors**" as the variant result. Please note that the difference between the "**u type**" residuals of a regression and the "**w type**" residuals of the same regression are explained in the **Reference** and **Model Type** manuals. For the purpose of this tutorial, let us just say that the major difference between these two types of regression errors is that the "**u type**" associated with variant "**sch_bc1auhg**" do not take in consideration the heteroskedastic and the autoregressive structures included in the variant, and that the "**w type**" do take these structures into account.

Again select the "**Scatter graph**" option in **module 4.10** and enter the title "**Heteroskedasticity Analysis**".

```
Select: Graph type
        1 = Horizontal line graph
        2 = Vertical   line graph
        3 = Scatter         graph
        4 = Bar             graph
        5 = Pie             graph
          3

Enter: Graph title  = Heteroskedasticity Analysis

Enter: Character size [1 - 6: default = 2] =

Define X axis values:

Select: Type of values
        1 = level variables
        2 = level variant results
          1

Enter: Algebraic expression = tend

Enter: X axis description (SECONDARY SCHOOL REGISTRATION) = Secondary
school registration

Enter: Character size [1 - 6: default = 2] =

Define Y axis values:
```

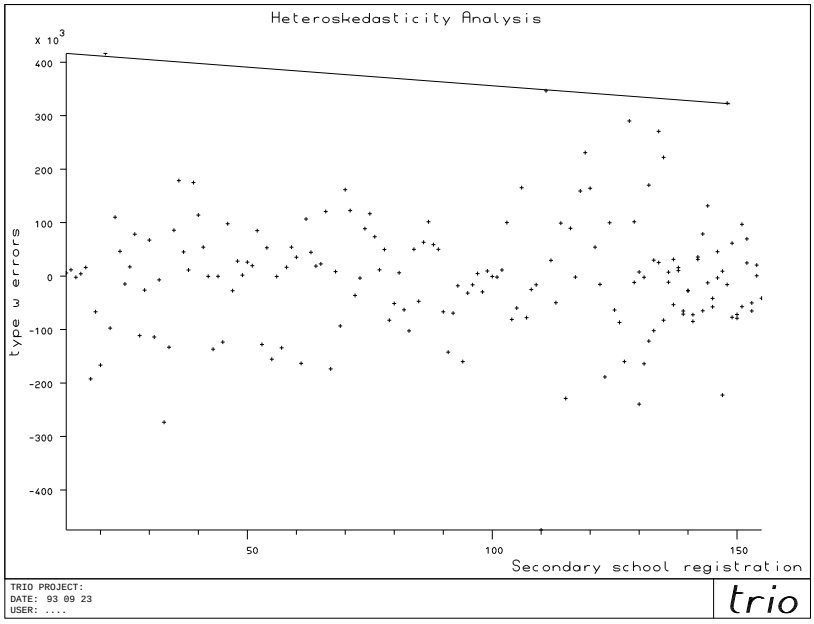
```
Select: Type of values
        1 = level variables
        2 = level variant results
          2

Enter: Variant name [:version] to consider = sch_bc1auhg

Select: Values
        1 = type u residuals (regression errors)
        2 = type u residuals in % vs a variable
        3 = type v residuals (homoskedastic errors)
        4 = type v residuals in % vs a variable
        5 = type w residuals (final model errors)
        6 = type w residuals in % vs a variable
        7 = E(Y): expected values of Y
        8 = Probability of Y not to be at the limit
        9 = E(Y**): expected values of transformed Y
       10 = Standard error of Y
       11 = Maximum likelihood forecast of E(Y)
       12 = Simulation forecast of E(Y)
       13 = Standard error of forecast
          5

Using estimated results? y

Enter: Y axis description (type w errors) =

Enter: Character size [1 - 6: default = 2] =

X Y plot is defined for period   13 to   181

Enter: First period to consider =

Enter: Last period to consider =

Define a constraint variable? n

Save in the temporary list of graphs? y

Plot? y
```

The following graph should appear on your screen. As you can notice, the "**type w errors**" are distributed more evenly than the "**type u errors**" around the zero value and the standard deviation of the **w type errors** is noticeably smaller than that associated with the "**type u errors**".

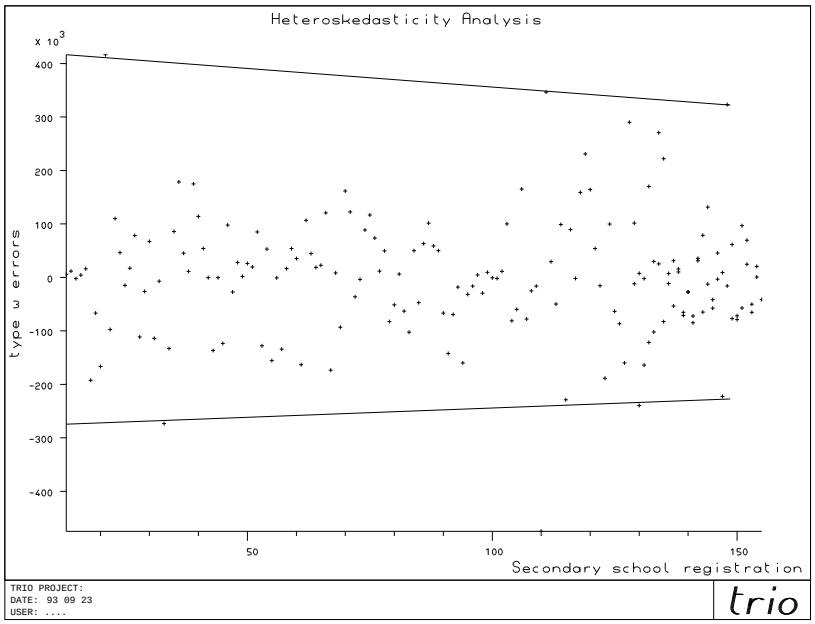


As we did previously, we will trace a line joining the extreme points of the scatter graph. Again press the key "**a**" to add an object and then the key "**p**" to select the polygon. Now you must select the first point of the object. Place the cursor on the Y axis around the value 420 and press the "**space bar**" to accept this first entry.

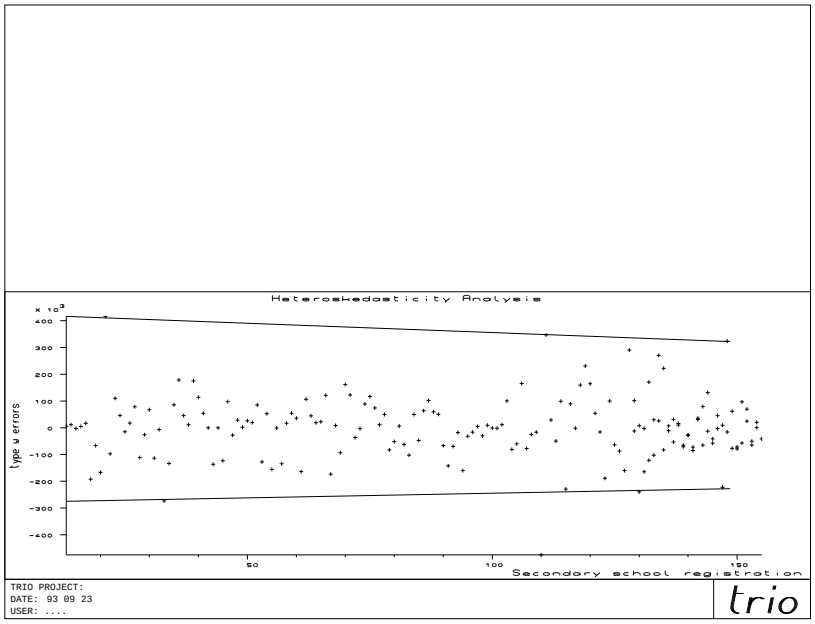
Select the last point of this first line by placing the cursor on the highest point at the right of the graph and then by pressing the key "**a**" to accept this entry. If you followed the above instructions you should get the following graph.



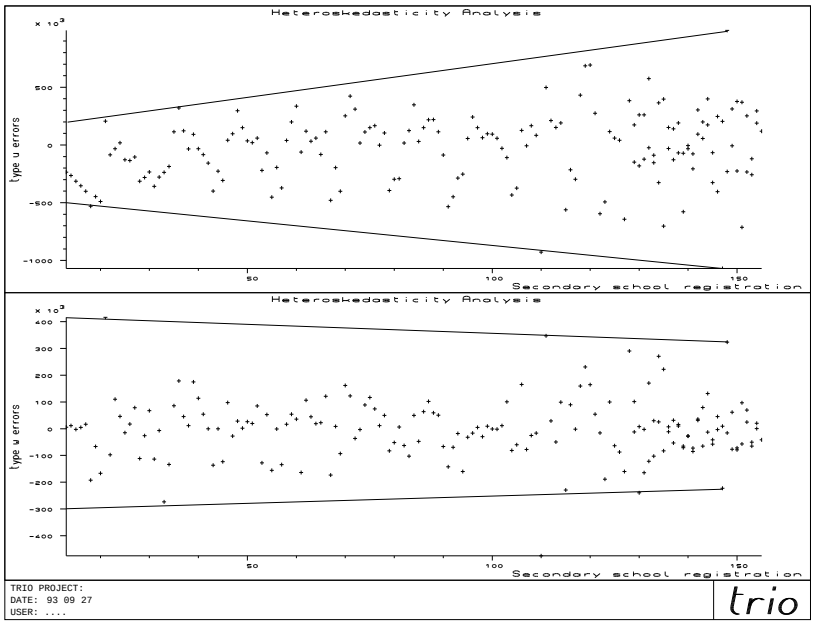
Proceed in the same manner in order to add the second line to the scatter graph. The first point of this new line should be on the Y axis somewhere around the value -300, and the second point should correspond to the lowest point on the right side of the scatter graph (Y = -215, X = 148). If you correctly selected these two last points, you should obtain the following graph.



We will now re-display the scatter graph but this time in the bottom part of the screen. To do this press the key "**v**" followed by the key "**d**" (for down). The result of these commands should be.



As a last graphic command to be learned during this tutorial, we will use the "**Z**" key to re-display all the active graphs. Remember that as long as you do not exit module 4.10 all the graphs that you create are active and can be re-displayed and even modified. Press the "**Z**" key and you should get the following graph.



Both graphs appear at the same time in a way that allows more easily the comparison between the two graphs, showing clearly in this example how the heteroskedastic and autoregressive structures correct the heteroskedasticity problem introduced in the regression by the variable "**tend**".

This presentation showed some of the advanced graphic capabilities of the TRIO package. For the rest of the graphic commands, the user should refer to Chapter 5 of the **Reference Manual**.

Appendix A
TRIO Word Editor Commands

Command letters	Action triggered by the command
Cursor movement	
d	Move one character to the right.
s	Move one character to the left.
wd	Move to the beginning of right word.
ws	Move to the end of left word.
x	Move to next field or line.
e	Move to previous field or line.
ld	Move to end of line.
ls	Move to beginning of line.
fd	Move to end of field or page.
fs	Move to beginning or field or page.
Insert and delete text	
i	Begin insertion of text; end insertion by typing ESC.
v	Overwrite text; end with ESC.
yy	Delete current character.
yw	Delete current word.
yl	Delete line or field.

Command letters	Action triggered by the command
Save and exit editing	
k	Save current page.
kq	Exit editing and save page.
q	Exit editing without saving page.
Display text	
z	Redisplay page.
n	Next page.
p	Previous page.
ml	Redefined left margin.
mr	Redefine right margin.
Block commands	
bb	Mark block begin.
be	Mark block end.
bc	Copy marked block at current cursor position.
bm	Move marked block at current cursor position.
by	Delete marked block.
bw	Write marked block to buffer.
br	Read marked block from buffer..
Miscellaneous commands	
h	Display help menu.

Appendix B
TRIO Editor Commands

SCROLLING COMMANDS	
Command letters	**Action triggered by the command**
Basic movement	
e	Scroll up
x	Scroll down
s	Scroll left
d	Scroll right
In page movement	
t	Top of page
b	Bottom of page
f	Move to first column
l	Move to last column
In file movement	
n	Next page
p	Previous page
N	Last page
P	First page
g*x*	Go to page *x*
Miscellaneous commands	
m	Mark current position
r	Return to marked position
h	Help
q	Quit

GROUP EDITOR COMMANDS	
Command letters	**Action triggered by the command**
Cursor movement	
e	Previous line
x	Next line
s	Previous field
d	Next field
Display text	
i	Insert at field
n	Insert after cursor
p	Insert before cursor
y	Delete field
z	Redisplay screen
Miscellaneous	
<	Permutation with previous variable
>	Permutation with next variable
h	Help
k	Save and quit
q	Quit no save

VARIANT EDITOR COMMANDS	
Command letters	**Action triggered by the command**
Cursor movement	
e	Previous line
x	Next line
s	Previous field
d	Next field
Insert and delete commands	
i	Insert at field
in	Insert a NO BC
ib	Insert a BC
i*x*	Insert a group (*x*) BC
*	Insert or remove a quasi-dummy
y	Delete at field
Miscellaneous	
z	Redisplay screen
k	Save and quit
q	Quit no save
h	Help

INITIALIZATION EDITOR COMMANDS	
Command letters	**Action triggered by the command**
Cursor commands	
e	Previous line
x	Next line
s	Previous field
d	Next field
Display text	
i	Insert at field
z	Redisplay screen
q	Quit no save
k	Save and quit
Miscellaneous	
h	Help
Parameters commands	
if	Insert fixed status for parameter
ie	Insert estimated status for parameter
in	Insert group number n

Appendix C
TRIO Graph Commands

Command letters	Action triggered by the command
Display	
z	Clear screen and display the current graph only.
Z	Clear screen and display all visible graphs.
k	Send a copy of the current graph to the plot file.
K	Send a copy of all visible graphs to the plot file
q	Quit graphic mode and return to primary select.
x	Display statistics
h	Help
Object editing	
p	Pick an object.
l	Display label.
m	Modify attributes.
c	Copy current object.
ad	Add a dot.
ar	Add a rectangle.
ac	Add a circle.
ap	Add a polygon.
at	Add a text string.
b#	Split interval in # bars.
bm	Modify the bar upper bound.
bs	Split current bar.
bj	Join bar with next bar.

Command letters	Action triggered by the command
Window editing	
w	Select window
wc	Move window center.
wl	Move window to the left.
wr	Move window to the right
wu	Move window up.
wd	Move window down.
w*n*	Resize *n* times.
wp	Get previous window.
w<space>	Define window.
Viewport editing	
v	Select viewport.
vl	Set viewport to the left part of the screen.
vr	Set viewport to the right part of the screen.
vu	Set viewport port to the upper part of the screen.
vd	Set viewport part to the lower part of the screen.
v*n*	Resize 1/*n* times.
vp	Get previous viewport.
v<space>	Define viewport.
Object transformation	
t	Translation
r	Rotation
s	Scaling
g	Global; transformation
u	Undo transformation
<.>	Repeat transformation

OTHER TRIO DOCUMENTATION

[CRT-441] • Babin, A., Dagenais, Florian, M., Gaudry, M., Guélat, J. , Lestage, P. and T. C. Liem "TRIO, An Open Interactive Graphic System for Demand Model Estimation", *Proceedings of the World Conference Transport Research 1986*, The Centre for Transport Studies, University of British Columbia, Or Publication #441, Centre de recherche sur les transports, Université de Montréal, January 1986 and March 1987.

Describes the general design concepts and the open computer based system, shows examples of graphic output. States that, in the transportation planning context, TRIO provides demand model calibration procedure which are complementary to the use of EMME/2.

[CRT-901] [AJD-101] • Gaudry, M. *et al.* "Cur Cum TRIO?" Publication #901, Centre de recherche sur les transports, Université de Montréal, April 1993.

Summarizes in English, French and German the professional, scientific and technical features of TRIO, notably the full integration of the four basic tasks required for regression work: the management of information, the production of models, the analysis of sata or model results, and assistance in report generation. Shows examples of text, TABLEX model results and graphic output.

[CRT-902] [AJD-102] • Gaudry, M., Lestage, P., Guélat, J. and P. Galvan, "TRIO Tutorial Version 1.0", Publication #902, Centre de recherche sur les transports, Université de Montréal, April 1993.

Conducts the user through a first session with TRIO. Principal steps covered include: creation of a database, model estimation with LEVEL, SHARE and PROBABILITY algorithms, use of the TABLEX table to examine regression results and production of graphs.

[CRT-903] [AJD-103] • Gaudry, M., Duclos, L.-P., Dufort, F. and T. C. Liem, "TRIO Reference Manual, Version 1.0", Publication #903, Centre de recherche sur les transports, Université de Montréal, April 1993.

Provides full description of all functions available in TRIO, as they pertain for instance to the system environment, the database, the variable and variant editors for all model classes, the analysis of data and model results, through tabular and graphical methods.

[CRT-904] [AJD-104] • Gaudry, M., Dagenais, M., Laferrière, R. and T. C. Liem, "TRIO Model Types, Version 1.0", Publication #904, Centre de recherche sur les transports, Université de Montréal, April 1993.

Contains a summary of the regression theory and networks used in TRIO. Demonstrates the properties and usefulness of the methods with selected applications. Describes in detail all available model types or significations, as well as the estimation techniques, which are programmed in TRIO.

[CRT 905] • Gaudry, M., Mandel, B. and U. Blum, "TRIO Tips for Beginners in Regression Analysis", Publication #905, Centre de recherche sur les transports, Université de Montréal, Forthcoming in 1993.

Will present regression analysis for non-technical beginners, with a strong emphasis on the intuitive understanding of the graphical, mathematical and statistical dimensions of regression analysis – with examples studied with TRIO.