Some implementation aspects of a Discovery System

Willi Klösgen
German National Research Center for Computer Science (GMD)
kloesgen@gmd.de

Abstract

Explora supports *Discovery in Databases* by large scale search for interesting instances of statistical patterns. Due to the variety of patterns and the immense combinatorial possibilities in studying relations between variables in subsets of data, at least two implementation problems arise. First, the user must be saved from getting overwhelmed with a deluge of findings. This can be achieved by some basic organization principles included into search. One principle is, to organize search hierarchically and to study first the strongest hypotheses (that mostly means the most general ones). Weaker hypotheses are then eliminated from further search. But even in moderately sized data, that approach alone usually does not prevent from large sets of findings. Therefore, in a second evaluation phase, a refinement strategy selects the most interesting verified statements and treats also the overlapping problem (due to correlations between independent variables). Further, the user can focus a discovery task by a more detailed specification of the analysis problem to be treated.

Second, it is important for discovery systems, to manage the efficiency problem. Each hypothesis evaluated when processing the large search space refers to subsets of cases stored in a database. These subsets correspond to combinations of variables and their (taxonomical) values. In principal, each subset needs random accesses to a lot of cases which takes much computation time.

We describe solutions implemented in the discovery system Explora to deal with these two problems. In an appendix, results of a discovery session in Explora are presented, and the necessity to insert more statistical strategies into a "higher" discovery level is discussed. On this level, instances of patterns verified during basic search are selected, refined, and combined to achieve a higher quality of presented findings including more interpretation potentiality.

1. Introduction

The rapid growth in number and size of existing and expected databases requires new tools and systems supporting the partial automation of data exploration and, as a final goal, of data comprehension.

In the last years, a new research area has been established offering first practical solutions in large scale data exploration. *Discovery in Databases* can be defined as the nontrivial extraction and high-level presentation of interesting information from data (Frawley, Piatetsky-Shapiro, & Matheus 1991). This new area (compare also: Zytkow 1992) benefits from tools and methods of Machine Learning, Statistics, Intelligent Data Bases, Knowledge Acquisition and Data Visualization.

The discovery system Explora (Hoschka & Klösgen 1991, Klösgen 1992a, Klösgen 1993) supports the discovery of interesting findings and unknown relations in databases by searching for instances of statistical patterns. A *pattern* is defined as a statement type (schema or model of a statement) by Frawley, Piatetsky-Shapiro, & Matheus (1991). An *instance* of a pattern is a statement in a high-level language that describes interesting information (a finding) in data. Patterns shall be understood directly by the (high-level) users of a discovery system. Discovery of findings in data means that we are searching in spaces of hypotheses for all instances of selected patterns that are interesting enough, according to some criteria measuring the degree of interestingness.

The most general pattern is a regularity (Zytkow & Baker 1991). The authors consider a linguistic representation of a regularity as a statement in a language based on the variables of a database and their values which specifies more (resp. less) probable events in the Cartesian product of the sets of values of the variables. Such a regularity has a *range* (which is some subset of the space of all events, defined by a logic expression) and a statistical relation describing for the range the probability of events. In the system Forty-Niner (Zembowicz & Zytkow 1992), statistical relations in form of contingency tables and equations are investigated.

In Explora, we mainly look for patterns describing *subgroups* of cases with outstanding distributional characteristics of the dependent variables (also goal variables, response variables, right hand sides of rules, etc.). This corresponds to events that are more, resp. less probable. Subgroups are constructed referring to the independent variables (also control variables, explanatory variables, left hand sides of rules, etc.). Subgroups are the primary search dimension for dependency patterns in Explora.

To describe the dependency between a dependent and some independent variables, Explora evaluates and lists subgroups of cases, for which e.g. the mean (continuous dependent variable) or the share (binary dependent variable) is extraordinarily high (low) within a population (or range).

The main dimensions of a typology for dependency patterns offered in Explora (Klösgen 1992b) are the number of populations compared in a pattern, the type of variables, deterministic or probabilistic verification methods, and the kind of language used to form subgroups, target groups, ranges, and populations of cases.

Explora constructs hierarchical spaces of hypotheses, organizes and controls the search for interesting instances in these spaces, verifies and evaluates the instances in data, and supports the presentation, management and outlining of the discovered findings. The variety of instances due to combinations of variables and populations (subsets of cases in which patterns appear) results in very large hypotheses spaces. A systematic, but not exhaustive search cuts away whole subspaces, without to skip important hypotheses.

Explora for the Apple-Macintosh™ running under MCL™ (Macintosh Common LISP) is free available (e.g. by "anonymous ftp". Open a connection to "ftp.gmd.de" and transfer the file "Explora.sit.hqx" from the directory "gmd/explora". The file "READ-ME" informs about the installation of Explora.) This version comes out in two ways. The system is available both for practical applications on medium sized data bases (for the Macintosh version upto 100.000 records) by end users, and for discovery research especially in the area of pattern construction, search strategies and high level presentation of findings. The version to be used for discovery research requires a user with an own MCL installation. This user may implement extensions and modifications of Explora in MCL. Therefore, this user must have an own license agreement with Apple Computer about MCL. The "end user" version of Explora is a stand alone program, distributed in object code and stripped of the access to LISP, not requiring a MCL license.

In this paper, we deal with two implementation problems for discovery systems. At first, we introduce into that part of the user interface of Explora offering the user to focus the discovery. Another aspect of a user interface for discovery systems is treated in the appendix. There, composing the results of an example session in Explora, the problem is addressed, how a discovery system could present its findings. As a second main point of this paper, the efficiency problem is discussed, explaining approaches of Explora for efficient data management and computation.

2. Focus of a Discovery Run

To focus a discovery run, the user gives a more detailed specification of an analysis problem. Two dialog components are available for that purpose: a focus window and a pattern menu.

The *focus window* holds areas for the selection of variables to be used as dependent, resp. independent variables, the selection of variables to be used for the construction of populations (ranges), the detail specification of the language used to form subgroups, target groups, ranges, and populations of cases, and the modification of parameters used by the verification method of a pattern.

The *pattern menu* offers a list of patterns. Patterns of a first set of patterns available in this menu compare a subgroup of a population with the population as a whole. For a second set of patterns, a subgroup is outstanding when compared in two populations. For patterns of a third set, a subgroup is outstanding when compared in *k* (*k*>2) populations. In each set, different patterns are available according to the variable type of the dependent variables. This arrangement is similar to general method-finding tables used in statistics (Koopmans 1981) which rely on a combination of problem type (one-sample, two-sample, *k*-sample comparison) and variable type.

2.1 The pattern menu

When a pattern is chosen in the pattern menu, the active focus window (Figure 1) is updated. Selections of variables remain valid as far as possible. However, depending on the pattern, variables of special types are admitted for selection in the lists of variables.

The general form of a pattern available in version 1.0 of Explora is determined by:

(1) *Distribution of dependent variables is outstanding for subgroup.*

The primary search dimension is given by a space of subgroups (constructed according to the selection of discrete independent variables, their taxonomies, and their combination options; compare 2.2). This search dimension builds the "inner loop" for the ordering of instances of patterns, i.e. subgroups are varied first, while the other arguments of a pattern still remain constant. For most patterns, this argument holds the redundancy-filter "True --> successor not interesting" (compare: Hoschka & Klösgen 1991). Statements are presented for as general subgroups as possible (still satisfying the evaluation criteria). If a statement is presented for a subgroup, then the subspace of the more special subgroups is cut from further search.

For patterns of the first set, (1) is specialized by the following definition of "outstanding":

1.1 *A subgroup of a population is outstanding, if the distribution of the dependent variable(s) in the subgroup differs significantly from the distribution in the population.*

A population is given by a range of a segment.

A discovery runs in one or several *segments* which are distinguished subsets of all cases of the database (for instance, the segment of all cases belonging to a special time point; compare 3.1 for the introduction of segments). A *range* is a subset, defined by some logical selection criterion (e.g. males, or young persons living in North). E.g., the subset "males in 1990" is a population.

The second search dimension is given by a space of ranges (constructed according to the selection of discrete variables in the selection list "Ranges", comp. 2.2). This search dimension holds the redundancy-filter "True --> successor not interesting".

The third search dimension refers to a set of segments (chosen in the selection list "Segments"; comp. 2.2). Segments are introduced to support efficient data access and flexible data structures.

The fourth search dimension is given by a set of "dependent variables". According to the type of these variables, further specializations are made. This is the "outer loop" for the ordering of statements (pattern instances), i.e. this argument varies finally.

A first pattern of this set is the "Dichotomy" pattern. For the fourth search dimension of this pattern (set of dependent variables), a space of target groups is constructed according to the selection of discrete variables in the selection list "Dependent variables" (comp. 2.2). A ***target group*** is a subset defined by a selection criterion referring to dependent variables and their (taxonomical) values.

The dichotomy "target group" versus "complement of target group" is analysed. The fourth search dimension holds the redundancy-filter "True --> successor not interesting". The verification method analysing a 2X2 contingency table uses a statistical test described in (Klösgen 1992a) to identify an outstanding subgroup:

1.1.1 *Share of target group is significantly larger/smaller in subgroup than in population.*

With a next pattern, strong sufficient rules (for a target group) of the following kind are discovered:

1.1.2 *Within population: If case belongs to subgroup, then case belongs to target group.*

The attribute "strong" means, that a percentage of cases is given the rule must at least be valid for ("exact" rules: 100%). Strong necessary rules (for a target group) relate to another pattern:

1.1.3 *Within population: If case belongs to target group, then case belongs to subgroup.*

This pattern holds the redundancy-filter "True --> predecessor true" for the argument "subgroup". Therefore, subgroups as special as possible are searched.

A "Discrete Distribution" pattern analyses the full contingency table for the *n* values of the dependent variable which exist in the population. Only one discrete dependent variable can be selected. The verification method uses a chi2 test to evaluate the significance of this 2X*n* contingency table:

1.1.4 *Distribution of dependent variable in subgroup differs significantly from distribution in population.*

The next and the following patterns require the selection of one continuous dependent variable. A mean-pattern holds the following substance:

1.1.5 *Mean of dependent variable is significantly larger/smaller in subgroup than in population.*

A statistical mean test is applied to verify an instance of this pattern. Another, elementary version of this pattern requires for a subgroup, that the mean is at least 10% (as default parameter) higher, resp. lower, than the mean in the population. Applying another similar pattern, one searches for subgroups with an overproportional cumulated value of a continuous variable. The cumulation in the subgroup relative to the cumulation in the population is overproportional, if compared with the size of the subgroup relative to the size of the population.

A preliminary, still elementary version of a "Median" pattern analyses the median of one continuous dependent variable, and a "Subpopulation-share" pattern is frequently applied for the analysis of market shares, e.g. the analysis of the market share of Product B in the coffee market. The dependent variable is a continuous variable (e.g. Price, Weight). Then the market share of sales of Product B (measured in prices or weights) in a population of cases (e.g. single coffee-sales in supermarkets) is analysed.

For patterns of the second subset, the term "outstanding" is defined in the following way:

1.2 *A subgroup is outstanding, if the distribution of the dependent variable(s) in this subgroup of a first population is significantly different from the distribution in this subgroup of a second population.*

The subgroup is compared in two populations. According to the type of the dependent variables, similar specializations are offered as for the first subset of patterns (1.1.*i*).

The main difference between the patterns of the third subset and the patterns for two populations (second subset) is the following. Comparing two populations is reduced to comparing two numbers with the specializations: "first number is significantly larger than second number" and "first number is significantly smaller than second number". These opposite cases are also important for some refinement techniques. Comparing *k* numbers leads to the more general statement "*k* numbers are significantly different", including also the case *k* = 2.

Within this subset, the distribution of dependent variables in a subgroup of a population is compared for k populations. A subgroup is outstanding, if the k distibutions (for the k populations) are significantly different.

The statistical patterns can be combined with elementary patterns, searching e.g. for regularities in the contingency tables underlying the statistical test of a pattern. For the dependent variable, the independent variable, and the formation of populations, we distinguish the nominal, ordinal, and time-oriented type. Then elementary patterns identify, respectively, ranking, monotonic, and time-series patterns in the tables. These patterns can be defined by heuristic criteria or by further statistical tests. Some examples of ranking patterns are: one value is distinctly the no.1, there is a leading group of several values; monotonic patterns are: monotonic, semi-monotonic, edge-centered, convex, concave; simple time-series patterns are: "the best value since ...", "*n* successive increases", etc.

2.2 The Focus Window

Before starting a discovery run, the user can select segments, ranges, and variables to be used for the active pattern, assign parameters for the verification method of the pattern, and set options for the combination of variables. This is done in the focus window. The language for the construction of groups of cases is a further aspect of a pattern typology. Expressions of propositional or predicate logic can be used to describe a subset of cases. Propositional logic uses the variables (attributes) and their taxonomical values (attributive language). Expressions in predicate logic rely on predicates. For databases, we have unary predicates (corresponding to variables and their taxonomical values), and n-ary predicates ($n > 1$) which mainly connect the relations (subfiles) of a database.

In the simplest propositional case, only conjunctions of order n (i.e. at most n conjunctions) are formed. To restrict the number of internal disjunctions of values of a variable (e.g. all intervals built with the values of an ordinal variable, or all internal disjunctions with the values of a nominal variable), one can define a hierarchical structure (taxonomy) holding only those nodes which correspond to internal disjunctions being of interest in the application domain.

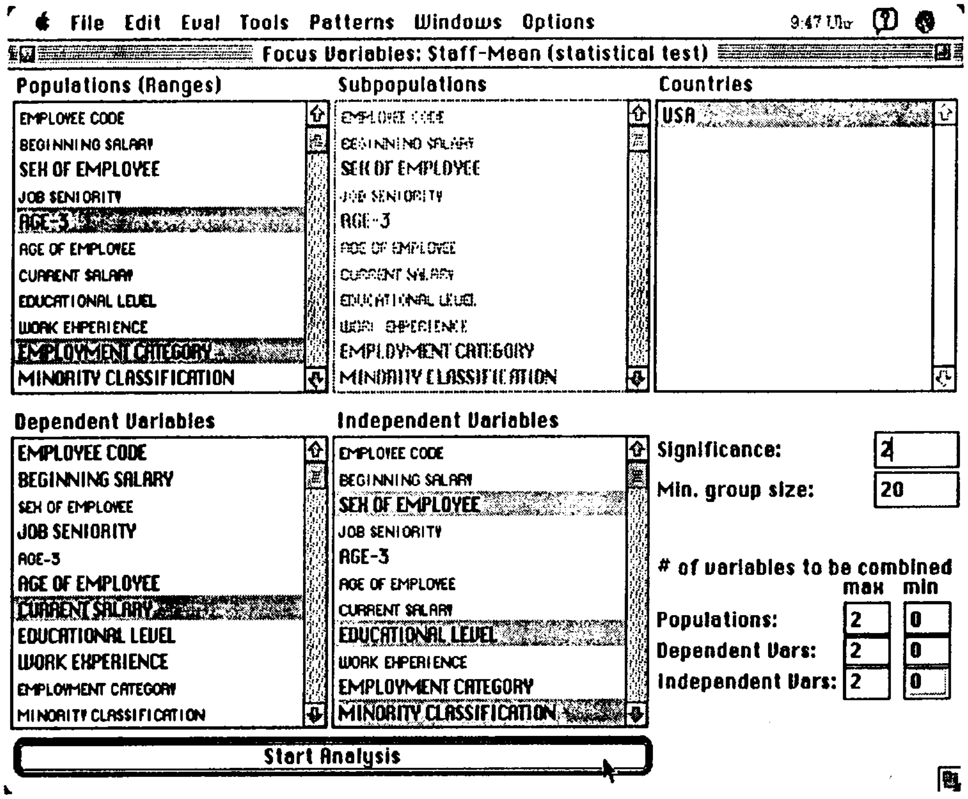
Depending on the language used for the formation of subgroups, target groups, and ranges, the search spaces can become very large. To restrict these spaces - for efficiency reasons and also to focus the search on the concepts of interest - the user of Explora has the possibility to compose the elements of a domain dependent sublanguage. This can be done for an individual discovery run by selecting the independent variables, choosing subsets of the elements of taxonomies or subsets of intervals for these variables, restricting the conjunctions, etc. In the version 1.0 of Explora, only conjunctions of variables are possible.

Figure 1 shows a focus window for the pattern "Mean: 1 population (statistical test)" and the corresponding results. This discovery runs in personal data (staff data) holding variables on the employees of a fictitious multi-national company. For this application ("Staff"), countries were introduced as segments (some variables may differ for countries). Only one segment is active (comp. 3.1), therefore only data on USA-Staff are evaluated (and read into main memory) for this discovery session.

Selection lists for variables

The first selection list for variables (upper left) relates to the ranges, for which instances of the active pattern are searched. Only discrete variables can be selected here. According to the specifications, a set of ranges is constructed.

In the example, the variables AGE-3 (a three-element discretization of the continuous variable AGE) and EMPLOYMENT CATEGORY were selected. Since the combination option for this selection list (lower right area in focus window) allows 0 to 2 combinations, the set of ranges includes "All employees" (combination of 0 variables), 3 age groups, 7 employment categories, and the 21 combinations of age and employment categories. This set is partially ordered: "All employees" is more general than "AGE>40" which is more general than "AGE>40, CLERICAL".



Focus Variables: Staff-Mean (statistical test)

Populations (Ranges)
EMPLOYEE CODE
BEGINNING SALARY
SEH OF EMPLOYEE
JOB SENIORITY
AGE-3
AGE OF EMPLOYEE
CURRENT SALARY
EDUCATIONAL LEVEL
WORK EHPERIENCE
EMPLOYMENT CATEGORY
MINORITY CLASSIFICATION

Subpopulations
EMPLOYEE CODE
BEGINNING SALARY
SEH OF EMPLOYEE
JOB SENIORITY
AGE-3
AGE OF EMPLOYEE
CURRENT SALARY
EDUCATIONAL LEVEL
WORK EHPERIENCE
EMPLOYMENT CATEGORY
MINORITY CLASSIFICATION

Countries
USA

Dependent Variables
EMPLOYEE CODE
BEGINNING SALARY
SEH OF EMPLOYEE
JOB SENIORITY
AGE-3
AGE OF EMPLOYEE
CURRENT SALARY
EDUCATIONAL LEVEL
WORK EHPERIENCE
EMPLOYMENT CATEGORY
MINORITY CLASSIFICATION

Independent Variables
EMPLOYEE CODE
BEGINNING SALARY
SEH OF EMPLOYEE
JOB SENIORITY
AGE-3
AGE OF EMPLOYEE
CURRENT SALARY
EDUCATIONAL LEVEL
WORK EHPERIENCE
EMPLOYMENT CATEGORY
MINORITY CLASSIFICATION

Significance: 2
Min. group size: 20

# of variables to be combined	max	min
Populations:	2	0
Dependent Vars:	2	0
Independent Vars:	2	0

Figure 1: Example of a focus window

Results of this discovery:
(Parts of the contents of the result window were copied (Edit Menu) and pasted into this text file).

```
Pattern: Probabilistic rule (mean), continuous dependent variable

Population: Employees of OUR-COMPANY, USA.
Mean of the variable CURRENT SALARY in the population:      13768
The mean is larger in the groups:
    MALES                                                   16577
    EDUCATIONAL LEVEL 16                                    19290
    EDUCATIONAL LEVEL > 16                                  27141

Refinement:
    EDUCATIONAL LEVEL > 16, MALES, WHITE                    28251
    EDUCATIONAL LEVEL > 16, MALES                           27457
    EDUCATIONAL LEVEL 16, MALES                             21505
    EDUCATIONAL LEVEL 16, WHITE                             19736
    EDUCATIONAL LEVEL 16                                    19290

Population: AGE OF EMPLOYEE > 40, CLERICAL, OUR-COMPANY, USA.
Mean of the variable CURRENT SALARY in the population:       9422
The mean is larger in the groups:
    NONWHITE                                                 9892
```

In the discovery run, instances of the pattern "Mean" are at first evaluated for the range "All employees". Verified instances are presented in the result window, but no longer analysed for the sub-ranges (redundancy filter "True->Successor not interesting"). Therefore, results about subranges (e.g. AGE OF EMPLOYEE > 40, CLERICAL) are only presented, if they differ from the results about their superranges (e.g. All Employees).

The list "Dependent Variables" (lower left) is used to select the dependent variables (also goal variables, response variables, right hand sides for classification rules, etc.). For the pattern "Mean", the selection of one continuous variable is admitted in this list. In the example, the continuous variable CURRENT SALARY was chosen. However, what selection is allowed here, depends on the active pattern. E.g. for the pattern "Dichotomy", the selection of several discrete variables is admitted. Then a partially ordered set of target groups is constructed.

The list "Independent Variables" (lower middle) is used to select the independent variables (also control variables, explanatory variables, left hand sides for classification rules, etc.). A partially ordered set of subgroups is constructed in the same way as a set of ranges is constructed for the selection list of "Ranges" (described above).

In the example, the variables SEX, EDUCATIONAL LEVEL, MINORITY CLASSIFICATION were selected. Again the redundancy filter "True->Successor not interesting" (defined for the pattern "Mean") causes, that if the mean of the variable CURRENT SALARY is significantly higher in a subgroup (for instance "Males"), all successor subgroups (e.g. "Males, White") are cut away from further search.

Refinement of findings

In a refinement phase, methods described in (Gebhardt, 1991) in more detail are used to optimize the set of findings. The redundancy filters applied during basic search cut the search graph. Therefore, in basic search, all hypotheses in the subgraphs of the presented findings are excluded from further search. To automatically exploit also these subgraphs, optimization techniques are available. Their main goal is to choose a moderately sized subcollection of findings that are sufficiently different from one another. The idea is that the user is less interested in all findings (according to a given criterion), if they are quite similar rather than in some diverse ones, even if they are, taken individually, less satisfactory.

This vague goal concept is made more precise by a procedure that employs two notions: a measure for the quality of a single finding (called evidence) and an asymmetric measure for the similarity of two findings (called affinity). These two cooperate in suppressing findings that are worse than, but not too different from, another finding.

In the above refinement results, "Educational Level 16" is not suppressed by "Educational Level 16, White" or "Educational Level 16, Males". A necessary condition for not supressing the superset "Educational Level 16" is, that the complement of the subset ("Educational Level 16, Nonwhite") is also positive (mean: 15983; larger than average 13768). Also the mean in "Educational Level 16, Females" is positive (16061). However, the refinement algorithm suppresses "Educational Level > 16" by "Educational Level > 16, Males". This is due to the negative behaviour of "Educational Level > 16, Females" (mean: 12898). Because of the negative behaviour of "Educational Level > 16, White, Females" (mean: 11640), also the group "Educational Level > 16, White" is eliminated. Because all male groups with an educational level smaller than 16 hold a mean below the average, the group "Males" is suppressed too.

Therefore, the refinement algorithm eliminates the "incorrect" findings on "Males" and "Educational Level >16" and identifies the "correct" finding on "Males, Educational Level ≥ 16". Whereas the group "Educational Level > 16, Males" is strengthened by "Educational Level > 16, Males, White" (because there is a clear additional effect of "White"), the group "Educational Level 16, Males" is not strengthened by "Educational Level 16, Males, White" (because there is no distinct additional effect of "White"). This and other properties of the refinement algorithm are discussed in (Gebhardt, 1991). Compare also the user manual of Explora. Further results of this refinement algorithm referring to the problem of correlations between independent variables are shown in the appendix.

Further specifications in focus window

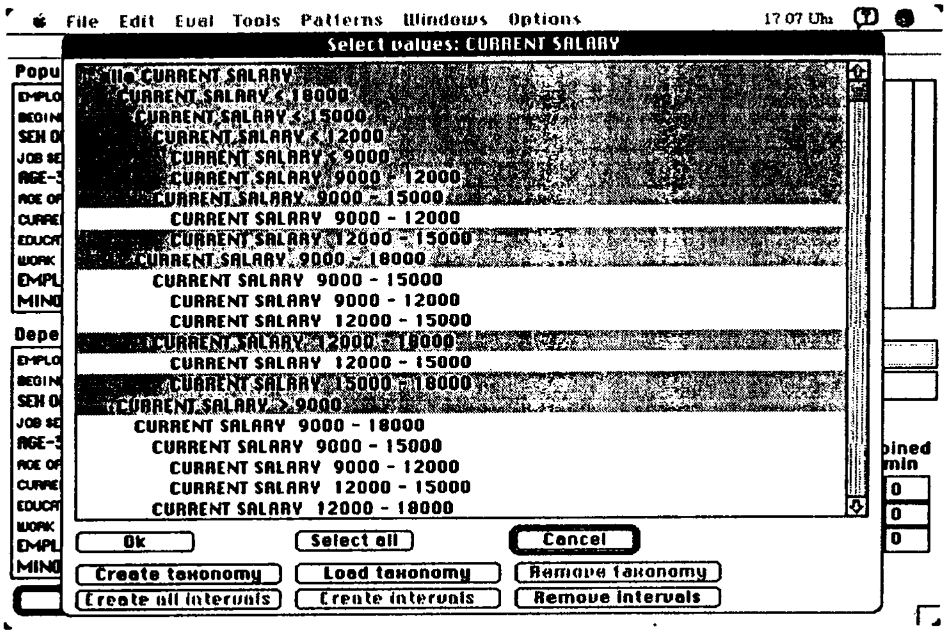
The list for selection of segments is placed in the upper right area of the focus window ("Countries" in this example). In case of an active pattern of the group "analysis of 1 population", the selected segments are used in sequence, that means, instances of pattern are searched at first in the first selected segment, then in the second segment, etc. In case of an active pattern belonging to the groups "comparison of 2, resp. *k* populations", 2 or *k* segments may be selected. Then these segments are compared (e.g. "USA" with "JAPAN").

Depending on the active pattern, a block of parameters for the verification method of this pattern appears in the middle right area of the focus window. The lower right block of combination options allows to restrict the number of variables to be combined, individually for the ranges (populations), dependent variables, and independent variables.

Taxonomies and intervals

The user can restrict the set of values of a discrete variable and can introduce taxonomies, or intervals for ordinal variables. Then only the selected values, resp. the taxonomies or intervals are included into the following discovery run, when ranges, target groups, or subgroups are built. After double-clicking into an entry of a selection list for variables (entry must be a discrete variable), a window for value selection is presented.

Using this window, also the generation of intervals for an ordinal discrete variable can be requested. The values can be selected which shall be used for the generation of intervals. Figure 2 shows, which intervals are generated when pressing the "Create all intervals" or "Create intervals" button. The basic ordinal variable is a 5-class discretisation of the variable "SALARY". In figure 2, all intervals were created. The leaves in this tree are the 5 original ordinal values (Salary < 9000, Salary 9000 - 12000, Salary 12000 - 15000, Salary 15000 -18000, Salary > 18000). The inner intervals (like 9000-18000) are missing, when only the boundary intervals were created.



```
All CURRENT SALARY
  CURRENT SALARY < 18000
    CURRENT SALARY < 15000
      CURRENT SALARY < 12000
        CURRENT SALARY < 9000
        CURRENT SALARY  9000 - 12000
      CURRENT SALARY  9000 - 15000
        CURRENT SALARY  9000 - 12000
        CURRENT SALARY  12000 - 15000
    CURRENT SALARY  9000 - 18000
      CURRENT SALARY  9000 - 15000
        CURRENT SALARY  9000 - 12000
        CURRENT SALARY  12000 - 15000
      CURRENT SALARY  12000 - 18000
        CURRENT SALARY  12000 - 15000
        CURRENT SALARY  15000 - 18000
  CURRENT SALARY > 9000
    CURRENT SALARY  9000 - 18000
      CURRENT SALARY  9000 - 15000
        CURRENT SALARY  9000 - 12000
        CURRENT SALARY  12000 - 15000
      CURRENT SALARY  12000 - 18000
```

Figure 2: All intervals of an ordinal variable were created.

3. Efficiency of Implementation

Explora is implemented in Common Lisp using object oriented programming techniques. Some advantages of object oriented techniques for discovery systems are mentioned in (Klösgen 1992). Now, we want to concentrate on efficiency requirements of Discovery Systems.

A verification method needs (for each segment) the following data for a single hypothesis:

- cases (in form of case-identifiers) belonging to a subgroup
- cases belonging to a target group (in case of discrete dependent variables) or all values of a continuous dependent variable
- cases belonging to a range

To collect the cases for one of these groups, those records in the database must be selected which hold special values for the variables combined in the group definition (e.g. old persons with high income in 1990). Thousands of hypotheses are tested for one discovery run. Even if a selection of records in a large data base needs only a few seconds, a single such discovery run will take hours. To arrive at run times of a few seconds for a single discovery run to be executed in dialog, an immense speeding up is necessary. We demonstrate in this chapter, how Explora overcomes this time problem of data base access, and other problems resulting from ressource shortages like computation time and main memory.

3.1 Efficient Data Management

The record-oriented organization of data base systems is transformed into an inverted data organization which uses variables as access criterium. The basic external storage structure contains all values belonging to one variable and one segment. To perform this transformation, data must be imported in Explora, before discovery can run there. When importing data, Explora produces this special structure for the data to be stored, enabling efficient access and computation during processing of large scale hypotheses spaces. Data are imported from the SPSS-Portable-File-Format (export format).

This format was selected, because it can be produced from most statistical packages (like SPSS or SAS). Also, SPSS has connections to several data base systems. The additional possibility of importing simple ASCII data files is prepared.

The connection of Explora with a data analysis system is useful, because of tasks that can already be performed in the data analysis system. First, the data analysis system can support data management, including data transformations, join operations, and generation of subsets of data. Second, in the analysis system, a more detailed statistical evaluation and graphical presentation of selected hypotheses can be considered.

Explora manages the imported data in *applications*. An application holds the data for several *segments*. A segment, for instance, can refer to a time point. In this case, data are collected regularly (monthly, yearly). In general, the segmentation of an application refers to one distinct field of the (original) data base. If, for instance, we have data about several countries, we could introduce different segments for different countries. When the user describes an application in Explora, especially he has to define the segments.

Segments can differ in structure, that means they can hold different variables (e.g. a new variable is available for a time-point). In this case, some restrictions must be observed when comparing segments.

When importing data, the user has to select the segment which shall receive the data. Data can be imported incrementally. One or several variables can be imported during one import procedure.

Segments are introduced in Explora to allow some flexibility in data structure (different variables in different segments, different value sets of a discrete variable for different segments) and to augment efficiency during discovery. When describing an application, the user can dynamically select active segments. Only the data for these segments are used (and read into main memory) during discovery.

Variables are the basic unit of data import. On a first level, Explora distinguishes *continuous* and *discrete* variables. Variables of type "string" as well as variables of type "number" with associated value labels (in SPSS) are imported as discrete variables into Explora. Variables of type "number" without value labels are imported as continuous variables.

For each segment, a continuos variable is stored as a vector of that real numbers which belong to the cases of this segment. For each segment, a discrete variable is stored as a list of binary vectors. For each value of the discrete variable, a binary vector is produced with a "1" for the cases of this segment holding this value.

These vectors of real numbers and lists of binary vectors are stored externally in binary form which is not only compact, but also does not require any transformations (from character to binary) before these numbers can be processed. A very efficient access technique can be implemented by applying the incremental compiling techniques of LISP. One needs only to compile (output) or load (input) such a list or vector (a file holding an assignment of the list or vector), which can be done dynamically.

The set of variables used for discovery consists of a subset of the fields available in the database and possibly some derived, additional fields (data transformations or join operations over the relations). For instance, an additional field can receive the results of a conceptual clustering algorithm, storing for each case the cluster to which it belongs. Such derivations of additional variables should be generated in the primary system (e.g. SPSS). Transformations of continuous into discrete variables and taxonomies like region, occupation, branch-structures can be defined in Explora for the imported variables and their values.

3.2 Efficient Computation

Discrete variables are used to construct groups of cases. Consider e.g. the variables "sex" (male, female) and the variable "family status" (single, married, divorced, widowed). The group "male" consists of all cases in the database (within a segment or a range of a segment) holding the value "male" for the variable "sex". Variables are combined in Explora by logical disjunctions to form e.g. the groups "male, married" or "female, widowed". These combinations can be done in a very simple and efficient way with logical operations on the bit-vectors. The bit-vector for the group "male, married" is the logical product ("Logand") of the bit vectors for males and married persons. The verification methods need also the number of cases belonging to a group. This calculation can also be performed efficiently by using another operation on bit vectors ("Logcount").

Explora manages the data for variables in main memory. Only when a variable is selected by the user for the first time during a discovery session, the values of the variable which belong to the active segments are read into main-memory. Therefore, only a task dependent part of the data is managed in main memory concentrating on selected segments and variables.

Other approaches to augment efficiency relate to search organization. Search has to be organized efficiently and must find the strongest valid hypotheses. Therefore, we have to use some kind of structure in the search space. Explora uses a general graph searching algorithm and a redundancy elimination technique (compare Klösgen 1992a). Redundancy filters are used to cut the search space. Generally, more general nodes (hypotheses) are evaluated first, and in case of a positive result, are presented and their subgraph is eliminated from further search. Refinement of a finding can be started by the user by request or automatically by the system.

The redundancy problem deals with the elimination of redundant statements. A statement is redundant with regard to another statement, if it is a logical or substantial consequence of the other statement. We have already seen logical redundancies when discussing the inclusion pattern ("target group is included in a subgroup"). This statement is redundant for a subset of the target group as well as for a superset of the subgroup. Other logical redundancies refer, for instance, to ordered patterns like ranks or lengths: the best result since 1950 is also the best result since 1960.

Substantial redundancies refer to heuristic criteria. If, for example, a mean of a variable deviates in a subgroup in a positive way from the total mean, then the mean is not necessarily or logically different in all subsets of this subgroup. But it is not interesting, to present also all the subsets with a positively different mean. Interesting, however, are then the subgroups with no or negative deviation. The latter subgroups can be reached in Explora by navigation commands ("exceptions") or in an automatic search option. If this is done automatically, more time is needed for search and more statements are presented.

In case of the filter "Statement about object true -> Statement about successor true", the search algorithm starts at the leaves, and if it finds a true statement, it looks for the strongest true statements compared to this one. Then it can eliminate all stronger statements because they are false and all weaker ones because they are redundant. In case of the filter "Statement about object true -> Statement about successor not interesting", it starts at the root, and if a true statement is found, then all successors are eliminated. When the specifier "predecessor" is involved, the inversed ordering of the graph is taken.

Zytkow and Zembowicz (1992) describe a refinement strategy based on a preliminary search for regularities and a subsequent regularity refinement phase. A "crude grain" search mechanism screens the hypotheses space and captures regularities in a simple, preliminary form, at a predefined level of significance and strength. Such a first search can be conducted fast, so that exhaustive screening is possible. Computationally expensive refinement techniques are applied selectively to explore the neighborhood of a node in the hypotheses space by applying hill climbing methods. Also more specific and complex patterns can be incorporated in the refinement phase.

In a refinement phase, one could also incorporate weights, which may be available in the database for the cases. Then a first search is done without the time consuming weighted calculations, and a subsequent local search is scheduled incorporating the weights. The refinement phase can also treat the overlapping problem.

The overlapping problem refers to overlapping classes of cases due to high correlations between fields of the database, e.g. the variables "age", "income", "job status" of a person. If the subgroup of persons with age over 65 shows an outstanding behavior, then surely also the subgroup of retired persons shows this behavior, because these subgroups are nearly identical. The refinement algorithm of Explora selects between such overlapping statements.

Conclusion

Version 1.0 of Explora was developed, to make a former prototype version (which was restricted for discovery in some special datasets) generally available for discovery in any statistical datasets. Explora 1.0 is appropriate both for practical applications on medium sized data bases (for this Macintosh version upto 100.000 records) and for discovery research especially in the area of pattern construction, search strategies and high level based presentation of findings. Discovery in Databases needs still a lot of practical applications, evaluations, and improvements, before advanced discovery products can be made available. In this sense, the application and evaluation of Explora shall advance the state of the art in Discovery. Therefore, comments are highly appreciated.

Explora offers some 20 patterns for discovery. Some of these pattern are still in a preliminary status, especially their verification method must be replaced by a more advanced or more appropriate statistical test. Some patterns may not deliver findings which can be interpreted in an easy and sensible way. Further (new) patterns may be more appropriate for large scale discovery. On a higher level of discovery and automation of the exploration process, it is necessary, to combine different patterns and to use the discovery runs now possible in Explora as building blocks of discovery macros.

All this further discovery research must be supported by experimental applications of available discovery systems. Test data sets can be analysed using exploratory statistical systems (e.g. with interactive graphical features) and the results of these analyses can be compared with the results of analyses performed independently in discovery systems based on systematic, large scale search in hypotheses spaces.

References

Frawley, W.J.; Piatetsky-Shapiro, G.; & Matheus, C.J. 1991. Knowledge Discovery in Databases: An Overview. In *Knowledge Discovery in Databases* eds. G. Piatetsky-Shapiro and W. Frawley, 1-27. Cambridge, MA: MIT Press.

Gebhardt, F. 1991. Choosing among Competing Generalizations. *Knowledge Acquisition* 3, 361-380.

Hoschka, P. & Klösgen, W. 1991. A Support System for Interpreting Statistical Data. In *Knowledge Discovery in Databases* eds. G. Piatetsky-Shapiro and W. Frawley, 325-346. Cambridge, MA: MIT Press.

Klösgen, W. 1992a. Problems for Knowledge Discovery in Databases and their Treatment in the Statistics Interpreter EXPLORA. *International Journal for Intelligent Systems* vol 7(7), 649-673.

Klösgen, W. 1992b. Patterns for Knowledge Discovery in Databases. In *Proceedings of the ML-92 Workshop on Machine Discovery* ed. Zytkow J., 1-10. Wichita, Kansas: National Institute for Aviation Research.

Klösgen, W. 1993 *Explora: A support system for Discovery in Databases, Version 1.0, User Manual.* Sankt Augustin: GMD.

Koopmans, L.H. 1981. *An Introduction to Contemporary Statistics.* Boston, MA: Duxbury Press.

Zembowicz, R. & Zytkow, J. 1992. Discovery of Regularities in Databases. In *Proceedings of the ML-92 Workshop on Machine Discovery* ed. Zytkow J., 18-27. Wichita, Kansas: National Institute for Aviation Research.

Zytkow, J. 1992. *Proceedings of the ML-92 Workshop on Machine Discovery* ed. Zytkow J., Wichita, Kansas: National Institute for Aviation Research.

Zytkow, J. & Baker, J. 1991. Interactive Mining of Regularities in Databases. In *Knowledge Discovery in Databases* eds. G. Piatetsky-Shapiro and W. Frawley, 31-53. Cambridge, MA: MIT Press.

Appendix: Results of an example session

Another aspect of the user interface of a discovery system relates to the presentation of findings. Explora uses a natural language form. Results are arranged as text entries in a result window which holds full editing possibilities. The user can edit the presented results, input own comments, reorganize results, copy other texts. The contents of a result window can be stored in an Explora-archive. Also the usual options of a File Menu (e.g. Print) are available.

Experiences with Explora show that it is sometimes difficult to express the substance of statistical relations in compact natural language form. Also, if the text of a finding holds several numbers distributed over a header and repetitive parts, the user may have difficulties in associating these numbers. Further, users familiar with the interpretation of tabulations may have difficulties with the natural language form.

An "Extended results" option delivers more numerical results (small tabulations belonging to a finding and its environment). This option should be extended by graphical presentations. Also for navigation, interactive graphics (as used in graphical data analyses systems) could provide a very useful exploratory progress.

The following results refer to employee records for 474 individuals hired between 1969 and 1971 by a bank engaged in Equal Employment Opportunity litigation (compare chapter 10.1 "Searching for Discrimination" in the SPSS Base Manual). The number of records in this database is very small for discovery purposes, especially for identification of significant subgroups. Also, the problem discussed below, is not a very typical discovery problem. To treat this problem more seriously, one needs more variables and more elaborate statistical methods. Nevertheless, we choose these data, because they are generally available (SPSS) to allow some comparisons of the following results with other approaches.

The relations between Employment Category, resp. Beginning Salary, and Education, Race, Sex, Age are analysed. We can expect a dependency of Employment Category from Educational Level and of Beginning Salary from Employment category. Now, the interesting question is, whether there exist any direct influences of Sex and Race on Employment Category and Beginning Salary. We should further expect dependencies of Educational Level from Age, Sex, Race. But in this special population of persons hired by a bank, these dependencies are not so obvious (as perhaps in the total population of all persons), because the bank hires persons for Employment Categories which may require special profiles of Education, Sex, Race, Age. Therefore, the first results illustrate the correlations of Education, Sex, Race, Age in this population: *(Below, some annotations are added in italics).*

```
Population: Employees hired 1969-1971, Our-Company.
```

Mean Pattern:
(refinement results; refinement also treats the problem of finding the "optimal" intervals)

```
Mean of the variable EDUCATIONAL LEVEL in the population:    13.5
The mean is larger in the groups:
   AGE OF EMPLOYEE  28 - 48, WHITE, MALES                    15.5
   AGE OF EMPLOYEE  28 - 48, WHITE                           15.2

The mean is smaller in the groups:
   AGE OF EMPLOYEE  > 48                                     11.4
   FEMALES                                                   12.4
```

Some details:

Dichotomy Pattern (probabilistic "necessary" rule):

```
Of the target-group <EDUCATIONAL LEVEL 8>:
   45% (Pop. 11%) are  AGE OF EMPLOYEE > 55, WHITE
   66% (Pop. 22%) are  AGE OF EMPLOYEE > 48
```

Equivalent Dichotomy Pattern (probabilistic "sufficient" rule):

```
11% of the population are <EDUCATIONAL LEVEL 8>. These are:
   47% of AGE OF EMPLOYEE > 55, WHITE
   34% of AGE OF EMPLOYEE > 48

Of the target-group <EDUCATIONAL LEVEL 12>:
   36% (Pop. 18%) are  FEMALES, AGE OF EMPLOYEE < 28
   67% (Pop. 46%) are  FEMALES
40% of the population are <EDUCATIONAL LEVEL 12>. These are:
   80% of FEMALES, AGE OF EMPLOYEE < 28
   59% of FEMALES

Of the target-group <EDUCATIONAL LEVEL 14/15>:
   52% (Pop. 28%) are  AGE OF EMPLOYEE 25 - 33, MALES
   89% (Pop. 70%) are  AGE OF EMPLOYEE 25 - 48
```

```
Of the target-group <EDUCATIONAL LEVEL 16>:
   29% (Pop.  7%) are  AGE OF EMPLOYEE 28 - 33, WHITE, FEMALES
   71% (Pop. 36%) are  AGE OF EMPLOYEE 28 - 40, WHITE
   90% (Pop. 56%) are  AGE OF EMPLOYEE 28 - 48
Of the target-group <EDUCATIONAL LEVEL 17/18>:
   95% (Pop. 42%) are  MALES, AGE OF EMPLOYEE 28 - 48
Of the target-group <EDUCATIONAL LEVEL 19+>:
   97% (Pop. 38%) are  MALES, WHITE, AGE OF EMPLOYEE > 28
```

Some exact rules (mostly obvious age restrictions):

```
more than 99% of the group <EDUCATIONAL LEVEL 14/15> are:
  AGE OF EMPLOYEE > 25                           (100%, 92% in pop.)
more than 99% of the group <EDUCATIONAL LEVEL 16> are:
  AGE OF EMPLOYEE 25 - 55                        (100%, 79% in pop.)
more than 99% of the group <EDUCATIONAL LEVEL 17/18> are:
  AGE OF EMPLOYEE > 28                           (100%, 78% in pop.)
more than 99% of the group <EDUCATIONAL LEVEL 19+> are:
  MALES, AGE OF EMPLOYEE > 28                    (100%, 50% in pop.)
```

The next results show, how refinement in Explora treats these correlations when "explaining" Employment Category. A simple taxonomy (Low/High Employment Category) was introduced.

```
82% of the population are <Low Employment Category>. These are:
     95% of FEMALES
     89% of AGE OF EMPLOYEE < 30
     90% of AGE OF EMPLOYEE > 40
    100% of EDUCATIONAL LEVEL < 14
     97% of EDUCATIONAL LEVEL 14/15
     96% of NONWHITE
Refinement:
    100% of EDUCATIONAL LEVEL < 14
     97% of EDUCATIONAL LEVEL 14/15
     86% of EDUCATIONAL LEVEL 16, NONWHITE, AGE OF EMPLOYEE > 30

Of the target-group <Low Employment Category>:
     62% (Pop. 51%) are  EDUCATIONAL LEVEL < 14
     30% (Pop. 26%) are  EDUCATIONAL LEVEL 14/15

Population: EDUCATIONAL LEVEL 16, Employees 1969-1971, Our-Company.
41% of the population are <Low Employment Category>. These are:
     86% of AGE OF EMPLOYEE > 30, NONWHITE
     67% of AGE OF EMPLOYEE > 30, FEMALES
but  20% of MALES, WHITE

Population: EDUCATIONAL LEVEL > 16, Employees 1969-1971, Our-Company.
12% of the population are <Low Employment Category>. These are:
    100% of MALES, AGE OF EMPLOYEE > 40, NONWHITE
but   3% of MALES, AGE OF EMPLOYEE > 30, WHITE

18% of the population are <High Employment Category>. These are:
     29% of MALES
     34% of AGE OF EMPLOYEE  30 - 40
     59% of EDUCATIONAL LEVEL 16
     88% of EDUCATIONAL LEVEL > 16
     22% of WHITE
Refinement:
     96% of EDUCATIONAL LEVEL > 16, MALES, WHITE
     90% of EDUCATIONAL LEVEL > 16, MALES
     65% of EDUCATIONAL LEVEL 16, WHITE
Of the target-group <High Employment Category>:
     50% (Pop.  9%) are  EDUCATIONAL LEVEL > 16, MALES, WHITE
     40% (Pop. 11%) are  EDUCATIONAL LEVEL 16, WHITE
```

Finally some results referring to Beginning Salary are shown. Again, refinement treats the correlations between the independent variables:

```
Mean of the variable BEGINNING SALARY in the population:   6806.4
The mean is larger in the groups:
   MALES                                                    8120.6
   AGE OF EMPLOYEE  > 30                                    7344.4
   EDUCATIONAL LEVEL 16                                     8935.4
   EDUCATIONAL LEVEL > 16                                  13077.8
   High Employment Category                                12103.1
Refinement:
   High Employment Category, MALES                         12651.1
   High Employment Category                                12103.1

The mean is smaller in the groups:
   FEMALES                                                  5236.8
   AGE OF EMPLOYEE < 30                                     5861.8
   EDUCATIONAL LEVEL < 13                                   5281.2
   Low Employment Category                                  5665.6
   NONWHITE                                                 5871.6
Refinement:
   Low Employment Category, FEMALES                         5100.3
   Low Employment Category, EDUCATIONAL LEVEL < 13          5247.3
   Low Employment Category                                  5665.6
```

Employment categories with Race as additional factor:

```
Population: CLERICAL, 1969-1971, Our-Company.
Mean of the variable <BEGINNING SALARY> in the population: 5733.9
The mean is larger in the groups:
   WHITE, EDUCATIONAL LEVEL 16                              7701.6

Population: OFFICE TRAINEE, 1969-1971, Our-Company.
Mean of the variable <BEGINNING SALARY> in the population: 5479.0
The mean is larger in the groups:
   WHITE, AGE OF EMPLOYEE > 30                              6328.1
   WHITE, MALES                                             6262.3
```

Comparison of opposite groups:

```
Comparison:                                              MALES vs. FEMALES
The mean of BEGINNING SALARY is larger for MALES:
  All Employees                                           8121 vs. 5237
  WHITE, AGE OF EMPLOYEE > 40, EDUCATIONAL LEVEL 13-16   13188 vs. 6310
  WHITE, AGE OF EMPLOYEE > 40                             9935 vs. 5217
  WHITE                                                   8638 vs. 5340
  NONWHITE, AGE OF EMPLOYEE  30 - 40                      7178 vs. 5090
But smaller for these groups:
  OFFICE TRAINEE, AGE OF EMPLOYEE > 30, EDUCATIONAL LEVEL 13-16
                                                          6225 vs. 7000

Comparison:                                              WHITE vs.NONWHITE
The mean of BEGINNING SALARY is larger for WHITE:
  All Employees                                           7069 vs. 5872
  EDUCATIONAL LEVEL 13-16, AGE OF EMPLOYEE > 40           9568 vs. 5714
  EDUCATIONAL LEVEL > 16                                 13424 vs. 9958
But smaller for these groups:
 EDUCATIONAL LEVEL < 13                                   5263 vs. 5330
  EDUCATIONAL LEVEL 8, AGE OF EMPLOYEE > 30               5185 vs. 5798
  EDUCATIONAL LEVEL 12, AGE OF EMPLOYEE < 30              4965 vs. 5053
  AGE OF EMPLOYEE > 40, CLERICAL                          5306 vs. 5472
```