IBM SPSS Conjoint 23

Product Information

This edition applies to version 23, release 0, modification 0 of IBM SPSS Statistics and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Introduction to Conjoint Analysis

Conjoint analysis is a market research tool for developing effective product design. Using conjoint analysis, the researcher can answer questions such as: What product attributes are important or unimportant to the consumer? What levels of product attributes are the most or least desirable in the consumer's mind? What is the market share of preference for leading competitors' products versus our existing or proposed product?

The virtue of conjoint analysis is that it asks the respondent to make choices in the same fashion as the consumer presumably does—by trading off features, one against another.

For example, suppose that you want to book an airline flight. You have the choice of sitting in a cramped seat or a spacious seat. If this were the only consideration, your choice would be clear. You would probably prefer a spacious seat. Or suppose you have a choice of ticket prices: $225 or $800. On price alone, taking nothing else into consideration, the lower price would be preferable. Finally, suppose you can take either a direct flight, which takes two hours, or a flight with one layover, which takes five hours. Most people would choose the direct flight.

The drawback to the above approach is that choice alternatives are presented on single attributes alone, one at a time. Conjoint analysis presents choice alternatives between products defined by sets of attributes. This is illustrated by the following choice: would you prefer a flight that is cramped, costs $225, and has one layover, or a flight that is spacious, costs $800, and is direct? If comfort, price, and duration are the relevant attributes, there are potentially eight products:

Table 1. Product choices specified by attribute combinations.

Product	Comfort	Price	Duration
1	cramped	$225	2 hours
2	cramped	$225	5 hours
3	cramped	$800	2 hours
4	cramped	$800	5 hours
5	spacious	$225	2 hours
6	spacious	$225	5 hours
7	spacious	$800	2 hours
8	spacious	$800	5 hours

Given the above alternatives, product 4 is probably the least preferred, while product 5 is probably the most preferred. The preferences of respondents for the other product offerings are implicitly determined by what is important to the respondent.

Using conjoint analysis, you can determine both the relative importance of each attribute as well as which levels of each attribute are most preferred. If the most preferable product is not feasible for some reason, such as cost, you would know the next most preferred alternative. If you have other information on the respondents, such as background demographics, you might be able to identify market segments for which distinct products can be packaged. For example, the business traveler and the student traveler might have different preferences that could be met by distinct product offerings.

The Full-Profile Approach

Conjoint uses the **full-profile** (also known as full-concept) **approach**, where respondents rank, order, or score a set of profiles, or cards, according to preference. Each profile describes a complete product or service and consists of a different combination of factor levels for all factors (attributes) of interest.

1

An Orthogonal Array

A potential problem with the full-profile approach soon becomes obvious if more than a few factors are involved and each factor has more than a couple of levels. The total number of profiles resulting from all possible combinations of the levels becomes too great for respondents to rank or score in a meaningful way. To solve this problem, the full-profile approach uses what is termed a **fractional factorial design**, which presents a suitable fraction of all possible combinations of the factor levels. The resulting set, called an **orthogonal array**, is designed to capture the main effects for each factor level. Interactions between levels of one factor with levels of another factor are assumed to be negligible.

The Generate Orthogonal Design procedure is used to generate an orthogonal array and is typically the starting point of a conjoint analysis. It also allows you to generate factor-level combinations, known as **holdout cases**, which are rated by the subjects but are not used to build the preference model. Instead, they are used as a check on the validity of the model.

The Experimental Stimuli

Each set of factor levels in an orthogonal design represents a different version of the product under study and should be presented to the subjects in the form of an individual product profile. This helps the respondent to focus on only the one product currently under evaluation. The stimuli should be standardized by making sure that the profiles are all similar in physical appearance except for the different combinations of features.

Creation of the product profiles is facilitated with the Display Design procedure. It takes a design generated by the Generate Orthogonal Design procedure, or entered by the user, and produces a set of product profiles in a ready-to-use format.

Collecting and Analyzing the Data

Since there is typically a great deal of between-subject variation in preferences, much of conjoint analysis focuses on the single subject. To generalize the results, a random sample of subjects from the target population is selected so that group results can be examined.

The size of the sample in conjoint studies varies greatly. In one report [1], the authors state that the sample size in commercial conjoint studies usually ranges from 100 to 1,000, with 300 to 550 the most typical range. In another study [2], it is found that smaller sample sizes (less than 100) are typical. As always, the sample size should be large enough to ensure reliability.

Once the sample is chosen, the researcher administers the set of profiles, or cards, to each respondent. The Conjoint procedure allows for three methods of data recording. In the first method, subjects are asked to assign a preference score to each profile. This type of method is typical when a Likert scale is used or when the subjects are asked to assign a number from 1 to 100 to indicate preference. In the second method, subjects are asked to assign a rank to each profile ranging from 1 to the total number of profiles. In the third method, subjects are asked to sort the profiles in terms of preference. With this last method, the researcher records the profile numbers in the order given by each subject.

Analysis of the data is done with the Conjoint procedure (available only through command syntax) and results in a utility score, called a **part-worth**, for each factor level. These utility scores, analogous to regression coefficients, provide a quantitative measure of the preference for each factor level, with larger values corresponding to greater preference. Part-worths are expressed in a common unit, allowing them to be added together to give the total utility, or overall preference, for any combination of factor levels. The part-worths then constitute a model for predicting the preference of any product profile, including profiles, referred to as **simulation cases**, that were not actually presented in the experiment.

---

1. Cattin, P., and D. R. Wittink. 1982. Commercial use of conjoint analysis: A survey. *Journal of Marketing,* 46:3, 44-53.

2. Akaah, I. P., and P. K. Korgaonkar. 1988. A conjoint investigation of the relative importance of risk relievers in direct marketing. *Journal of Advertising Research,* 28:4, 38-44.

The information obtained from a conjoint analysis can be applied to a wide variety of market research questions. It can be used to investigate areas such as product design, market share, strategic advertising, cost-benefit analysis, and market segmentation.

Although the focus of this manual is on market research applications, conjoint analysis can be useful in almost any scientific or business field in which measuring people's perceptions or judgments is important.

Chapter 2. Generating an Orthogonal Design

Generate Orthogonal Design generates a data file containing an orthogonal main-effects design that permits the statistical testing of several factors without testing every combination of factor levels. This design can be displayed with the Display Design procedure, and the data file can be used by other procedures, such as Conjoint.

Example. A low-fare airline startup is interested in determining the relative importance to potential customers of the various factors that comprise its product offering. Price is clearly a primary factor, but how important are other factors, such as seat size, number of layovers, and whether or not a beverage/snack service is included? A survey asking respondents to rank product profiles representing all possible factor combinations is unreasonable given the large number of profiles. The Generate Orthogonal Design procedure creates a reduced set of product profiles that is small enough to include in a survey but large enough to assess the relative importance of each factor.

To Generate an Orthogonal Design
1. From the menus choose:

   Data > **Orthogonal Design** > **Generate...**
2. Define at least one factor. Enter a name in the Factor Name text box. Factor names can be any valid variable name, except *status_* or *card_*. You can also assign an optional factor label.
3. Click **Add** to add the factor name and an optional label. To delete a factor, select it in the list and click **Remove**. To modify a factor name or label, select it in the list, modify the name or label, and click **Change**.
4. Define values for each factor by selecting the factor and clicking **Define Values**.

Data File. Allows you to control the destination of the orthogonal design. You can save the design to a new dataset in the current session or to an external data file.
- **Create a new dataset.** Creates a new dataset in the current session containing the factors and cases generated by the plan.
- **Create new data file.** Creates an external data file containing the factors and cases generated by the plan. By default, this data file is named *ortho.sav*, and it is saved to the current directory. Click **File** to specify a different name and destination for the file.

Reset random number seed to. Resets the random number seed to the specified value. The seed can be any integer value from 0 through 2,000,000,000. Within a session, a different seed is used each time you generate a set of random numbers, producing different results. If you want to duplicate the same random numbers, you should set the seed value before you generate your first design and reset the seed to the same value each subsequent time you generate the design.

Optionally, you can:
- Click **Options** to specify the minimum number of cases in the orthogonal design and to select holdout cases.

Defining Values for an Orthogonal Design

You must assign values to each level of the selected factor or factors. The factor name will be displayed after **Values and Labels for**.

Enter each value of the factor. You can elect to give the values descriptive labels. If you do not assign labels to the values, labels that correspond to the values are automatically assigned (that is, a value of 1 is assigned a label of 1, a value of 3 is assigned a label of 3, and so on).

Auto-Fill. Allows you to automatically fill the Value boxes with consecutive values beginning with 1. Enter the maximum value and click **Fill** to populate the values.

Orthogonal Design Options

Minimum number of cases to generate. Specifies a minimum number of cases for the plan. Select a positive integer less than or equal to the total number of cases that can be formed from all possible combinations of the factor levels. If you do not explicitly specify the minimum number of cases to generate, the minimum number of cases necessary for the orthogonal plan is generated. If the Orthoplan procedure cannot generate at least the number of profiles requested for the minimum, it will generate the largest number it can that fits the specified factors and levels. Note that the design does not necessarily include exactly the number of specified cases but rather the smallest possible number of cases in the orthogonal design using this value as a minimum.

Holdout Cases. You can define holdout cases that are rated by subjects but are not included in the conjoint analysis.

- **Number of holdout cases.** Creates holdout cases in addition to the regular plan cases. Holdout cases are judged by the subjects but are not used when the Conjoint procedure estimates utilities. You can specify any positive integer less than or equal to the total number of cases that can be formed from all possible combinations of factor levels. Holdout cases are generated from another random plan, not the main-effects experimental plan. The holdout cases do not duplicate the experimental profiles or each other. By default, no holdout cases are produced.
- **Randomly mix with other cases.** Randomly mixes holdout cases with the experimental cases. When this option is deselected, holdout cases appear separately, following the experimental cases.

ORTHOPLAN Command Additional Features

The command syntax language also allows you to:
- Append the orthogonal design to the active dataset rather than creating a new one.
- Specify simulation cases before generating the orthogonal design rather than after the design has been created.

See the *Command Syntax Reference* for complete syntax information.

Chapter 3. Displaying a Design

The Display Design procedure allows you to print an experimental design. You can print the design in either a rough-draft listing format or as profiles that you can present to subjects in a conjoint study. This procedure can display designs created with the Generate Orthogonal Design procedure or any designs displayed in an active dataset.

To Display an Orthogonal Design

1. From the menus choose:

    Data > **Orthogonal Design** > **Display...**

2. Move one or more factors into the Factors list.

3. Select a format for displaying the profiles in the output.

Format. You can choose one or more of the following format options:

* **Listing for experimenter.** Displays the design in a draft format that differentiates holdout profiles from experimental profiles and lists simulation profiles separately following the experimental and holdout profiles.

* **Profiles for subjects.** Produces profiles that can be presented to subjects. This format does not differentiate holdout profiles and does not produce simulation profiles.

Optionally, you can:

* Click **Titles** to define headers and footers for the profiles.

Display Design Titles

Profile Title. Enter a profile title up to 80 characters long. Titles appear at the top of the output if you have selected **Listing for experimenter** and at the top of each new profile if you have selected **Profiles for subjects** in the main dialog box. For **Profiles for subjects**, if the special character sequence *)CARD* is specified anywhere in the title, the procedure will replace it with the sequential profile number. This character sequence is not translated for **Listing for experimenter**.

Profile Footer. Enter a profile footer up to 80 characters long. Footers appear at the bottom of the output if you have selected **Listing for experimenter** and at the bottom of each profile if you have selected **Profiles for subjects** in the main dialog box. For **Profiles for subjects**, if the special character sequence *)CARD* is specified anywhere in the footer, the procedure will replace it with the sequential profile number. This character sequence is not translated for **Listing for experimenter**.

PLANCARDS Command Additional Features

The command syntax language also allows you to:

* Write profiles for subjects to an external file (using the OUTFILE subcommand).

See the *Command Syntax Reference* for complete syntax information.

Chapter 4. Running a Conjoint Analysis

A graphical user interface is not yet available for the Conjoint procedure. To obtain a conjoint analysis, you must enter command syntax for a `CONJOINT` command into a syntax window and then run it.

- For an example of command syntax for a `CONJOINT` command in the context of a complete conjoint analysis--including generating and displaying an orthogonal design--see .
- For complete command syntax information about the `CONJOINT` command, see the *Command Syntax Reference*.

To Run a Command from a Syntax Window

From the menus choose:

File > **New** > **Syntax...**

This opens a syntax window.

1. Enter the command syntax for the `CONJOINT` command.
2. Highlight the command in the syntax window, and click the Run button (the right-pointing triangle) on the Syntax Editor toolbar.

See the *Core System User's Guide* for more information about running commands in syntax windows.

Requirements

The Conjoint procedure requires two files—a data file and a plan file—and the specification of how data were recorded (for example, each data point is a preference score from 1 to 100). The plan file consists of the set of product profiles to be rated by the subjects and should be generated using the Generate Orthogonal Design procedure. The data file contains the preference scores or rankings of those profiles collected from the subjects. The plan and data files are specified with the `PLAN` and `DATA` subcommands, respectively. The method of data recording is specified with the `SEQUENCE`, `RANK`, or `SCORE` subcommands. The following command syntax shows a minimal specification:

```
CONJOINT PLAN='CPLAN.SAV' /DATA='RUGRANKS.SAV'
 /SEQUENCE=PREF1 TO PREF22.
```

Specifying the Plan File and the Data File

The `CONJOINT` command provides a number of options for specifying the plan file and the data file.

- You can explicitly specify the filenames for the two files. For example:

  ```
  CONJOINT PLAN='CPLAN.SAV' /DATA='RUGRANKS.SAV'
  ```

- If only a plan file or data file is specified, the `CONJOINT` command reads the specified file and uses the active dataset as the other. For example, if you specify a data file but omit a plan file (you cannot omit both), the active dataset is used as the plan, as shown in the following example:

  ```
  CONJOINT DATA='RUGRANKS.SAV'
  ```

- You can use the asterisk (*) in place of a filename to indicate the active dataset, as shown in the following example:

  ```
  CONJOINT PLAN='CPLAN.SAV' /DATA=*
  ```

  The active dataset is used as the preference data. Note that you cannot use the asterisk (*) for both the plan file and the data file.

Specifying How Data Were Recorded

You must specify the way in which preference data were recorded. Data can be recorded in one of three ways: sequentially, as rankings, or as preference scores. These three methods are indicated by the `SEQUENCE`, `RANK`, and `SCORE` subcommands. You must specify one, and only one, of these subcommands as part of a `CONJOINT` command.

SEQUENCE Subcommand

The `SEQUENCE` subcommand indicates that data were recorded sequentially so that each data point in the data file is a profile number, starting with the most preferred profile and ending with the least preferred profile. This is how data are recorded if the subject is asked to order the profiles from the most to the least preferred. The researcher records which profile number was first, which profile number was second, and so on.

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /SEQUENCE=PREF1 TO PREF22.
```

- The variable *PREF1* contains the profile number for the most preferred profile out of 22 profiles in the orthogonal plan. The variable *PREF22* contains the profile number for the least preferred profile in the plan.

RANK Subcommand

The `RANK` subcommand indicates that each data point is a ranking, starting with the ranking of profile 1, then the ranking of profile 2, and so on. This is how the data are recorded if the subject is asked to assign a rank to each profile, ranging from 1 to n, where n is the number of profiles. A lower rank implies greater preference.

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /RANK=RANK1 TO RANK22.
```

- The variable *RANK1* contains the ranking of profile 1, out of a total of 22 profiles in the orthogonal plan. The variable *RANK22* contains the ranking of profile 22.

SCORE Subcommand

The `SCORE` subcommand indicates that each data point is a preference score assigned to the profiles, starting with the score of profile 1, then the score of profile 2, and so on. This type of data might be generated, for example, by asking subjects to assign a number from 1 to 100 to show how much they liked the profile. A higher score implies greater preference.

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /SCORE=SCORE1 TO SCORE22.
```

- The variable *SCORE1* contains the score for profile 1, and *SCORE22* contains the score for profile 22.

Optional Subcommands

The `CONJOINT` command offers a number of optional subcommands that provide additional control and functionality beyond what is required.

SUBJECT Subcommand

The `SUBJECT` subcommand allows you to specify a variable from the data file to be used as an identifier for the subjects. If you do not specify a subject variable, the `CONJOINT` command assumes that all of the cases in the data file come from one subject. The following example specifies that the variable *ID*, from the file *rugranks.sav*, is to be used as a subject identifier.

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /SCORE=SCORE1 TO SCORE22 /SUBJECT=ID.
```

FACTORS Subcommand

The FACTORS subcommand allows you to specify the model describing the expected relationship between factors and the rankings or scores. If you do not specify a model for a factor, CONJOINT assumes a discrete model. You can specify one of four models:

DISCRETE. The DISCRETE model indicates that the factor levels are categorical and that no assumption is made about the relationship between the factor and the scores or ranks. This is the default.

LINEAR. The LINEAR model indicates an expected linear relationship between the factor and the scores or ranks. You can specify the expected direction of the linear relationship with the keywords MORE and LESS. MORE indicates that higher levels of a factor are expected to be preferred, while LESS indicates that lower levels of a factor are expected to be preferred. Specifying MORE or LESS will *not* affect estimates of utilities. They are used simply to identify subjects whose estimates do not match the expected direction.

IDEAL. The IDEAL model indicates an expected quadratic relationship between the scores or ranks and the factor. It is assumed that there is an ideal level for the factor, and distance from this ideal point (in either direction) is associated with decreasing preference. Factors described with this model should have at least three levels.

ANTIIDEAL. The ANTIIDEAL model indicates an expected quadratic relationship between the scores or ranks and the factor. It is assumed that there is a worst level for the factor, and distance from this point (in either direction) is associated with increasing preference. Factors described with this model should have at least three levels.

The following command syntax provides an example using the FACTORS subcommand:

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /RANK=RANK1 TO RANK22  /SUBJECT=ID
 /FACTORS=PACKAGE BRAND (DISCRETE) PRICE (LINEAR LESS)
          SEAL (LINEAR MORE) MONEY (LINEAR MORE).
```

- Note that both *package* and *brand* are modeled as discrete.

PRINT Subcommand

The PRINT subcommand allows you to control the content of the tabular output. For example, if you have a large number of subjects, you can choose to limit the output to summary results only, omitting detailed output for each subject, as shown in the following example:

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /RANK=RANK1 TO RANK22 /SUBJECT=ID
 /PRINT=SUMMARYONLY.
```

You can also choose whether the output includes analysis of the experimental data, results for any simulation cases included in the plan file, both, or none. Simulation cases are not rated by the subjects but represent product profiles of interest to you. The Conjoint procedure uses the analysis of the experimental data to make predictions about the relative preference for each of the simulation profiles. In the following example, detailed output for each subject is suppressed, and the output is limited to results of the simulations:

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /RANK=RANK1 TO RANK22 /SUBJECT=ID
 /PRINT=SIMULATION SUMMARYONLY.
```

PLOT Subcommand

The PLOT subcommand controls whether plots are included in the output. Like tabular output (PRINT subcommand), you can control whether the output is limited to summary results or includes results for each subject. By default, no plots are produced. In the following example, output includes all available plots:

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /RANK=RANK1 TO RANK22 /SUBJECT=ID
 /PLOT=ALL.
```

UTILITY Subcommand

The UTILITY subcommand writes a data file in IBM® SPSS® Statistics format containing detailed information for each subject. It includes the utilities for DISCRETE factors, the slope and quadratic functions for LINEAR, IDEAL, and ANTIIDEAL factors, the regression constant, and the estimated preference scores. These values can then be used in further analyses or for making additional plots with other procedures. The following example creates a utility file named *rugutil.sav*:

```
CONJOINT PLAN=* /DATA='RUGRANKS.SAV'
 /RANK=RANK1 TO RANK22 /SUBJECT=ID
 /UTILITY='RUGUTIL.SAV'.
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Index

Special characters

IBM ®

Printed in USA