Simulation Software - Integrated environments. Examples and review of some existing software popular and useful in the industry, e.g., Arena, AutoMod, Extend, Flexsim, Micro Saint, Pro Model, Quest,SIMUL8, WITNESS etc. Simulation using languages and environments like C++/Java/GPSS/SSF etc. Experimentation and Statistical-Analysis Tools: common features and relevant current products.

Simulation software is a program which uses set of mathematical formulas to process a real phenomenon into a model. Simulation software helps you predict the behavior of a system without actually performing that operation. It is used to evaluate a new design, diagnose problems with an existing design, and test a system under conditions that are hard to reproduce, such as a satellite in outer space.

Simulation software with real-time response is often used in gaming, but it also has important industrial applications. Advanced computer programs can simulate power system behavior, weather conditions, electronic circuits, chemical reactions, mechatronics, heat pumps, feedback control systems, atomic reactions, even complex biological processes.

To run a simulation, you need a mathematical model of your system, which can be expressed as a block diagram, schematic, statechart, or even code. The simulation software calculates the behavior of the model as conditions evolve over time or as events occur. Simulation software also includes visualization tools, such as data displays and 3D animation, to help monitor the simulation as it runs. Engineers and scientists use simulation software for a variety of reasons:

- Creating and simulating models is less expensive than building and testing hardware prototypes.
- You can use simulation software to test different designs before building one in hardware.
- You can connect simulation software to hardware to test the integration of the full design.

In addition to imitating processes to see how they behave under different conditions, simulations are also used to test new theories. After creating a theory of causal relationships, the theorist can codify the relationships in the form of a computer program. If the program then behaves in the same way as the real process, there is a good chance that the proposed relationships are correct.

An **integrated development environment** (**IDE**) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger. Some IDEs, such as NetBeans and Eclipse, contain the necessary compiler, interpreter, or both; others, such as SharpDevelop and Lazarus, do not.

The boundary between an IDE and other parts of the broader software development environment is not well-defined; sometimes a version control system or various tools to simplify the construction of a graphical user interface (GUI) are integrated. Many modern IDEs also have a class browser, an object browser, and a class hierarchy diagram for use in object-oriented software development.

Examples : 1) **OpNet Modeler/IT Guru :Open source(OPNET** is a discrete event, object-oriented, general purpose network simulator (commercial simulation package). **OPNET IT GURU** is a smaller

version of **OPNET** Modeler which is available at no costs under **OPNET** academic program.) OPNET is a company the develops network simulation and analysis software.

2) **Matlab/SIMULINK :** block diagram focus
 focus on scientific/technical applications
 rich set of Block sets/Toolboxes
Matlab: Matrix Laboratory

Simulink is a simulation and model-based design environment for dynamic and embedded systems, integrated with MATLAB. Simulink, also developed by MathWorks, is a data flow graphical programming language tool for modelling, simulating and analyzing multi-domain dynamic systems. It is basically a graphical block diagramming tool with customizable set of block libraries.It allows you to incorporate MATLAB algorithms into models as well as export the simulation results into MATLAB for further analysis.Simulink supports −

- system-level design
- simulation
- automatic code generation
- testing and verification of embedded systems

There are several other add-on products provided by MathWorks and third-party hardware and software products that are available for use with Simulink.

The following list gives brief description of some of them −

- **Stateflow** allows developing state machines and flow charts.
- **Simulink Coder** allows the generation of C source code for real-time implementation of systems automatically.
- **xPC Target** together with **x86-based real-time systems** provide an environment to simulate and test Simulink and Stateflow models in real-time on the physical system.
- **Embedded Coder** supports specific embedded targets.
- **HDL Coder** allows to automatically generate synthesizable VHDL and Verilog.
- **SimEvents** provides a library of graphical building blocks for modelling queuing systems.

3) **MathCAD :** equation-based worksheets includes symbolic programming
Mathcad is computer software primarily intended for the verification, validation, documentation and re-use of engineering calculations. The figure below is a blank worksheet.

SIMULATION WITH ARENA : The Arena modeling system from Systems Modeling Corporation is a flexible and powerful tool that allows analysts to creates animated simulation models that accurately represent virtually any system. The Arena Basis and Professional edition are offered by Rockwell Automation. It is general purpose simulation software that can be used for simulating discrete and continuous system. The Arena Basis edition is used for high level analysis with help flowcharts. The Arena Professional edition is designed for simulating discrete and continuous system.

EXAMPLE: A SINGLE COUNTER TRANSACTION

Customers arrive randomly: described by a distribution
Transacts business: single counter
Leaves
E.g.: an ATM counter

Modules

• Flow chart modules & Data modules
– Basic modules
• Create
• Process
• Dispose

SIMUL8 simulation software is a product of the SIMUL8 Corporation used for simulating systems that involve processing of discrete entities at discrete times. This program is a tool for planning, design, optimization and reengineering of real production, manufacturing, logistic or service provision systems. SIMUL8 allows its user to create a computer model, which takes into account real life constraints, capacities, failure rates, shift patterns, and other factors affecting the total performance and efficiency of production.[1] Through this model it is possible to test real scenarios in a virtual environment, for example simulate planned function and load of the system, change parameters affecting system performance, carry out extreme-load tests, verify by experiments the proposed solutions and select the optimal solution. A common feature of problems solved in SIMUL8 is that they are concerned with cost, time and inventory.

Simulation in java

• Java is a widely used programming that has been used extensively in simulation.
• The following components are common to almost all models written in java
• **Clock:** a variable defining simulated time
• **Initialization method: a** method to define the system state at time 0.
• **Min-time event methods:** a method that identifies the imminent event, that is the element of the future event list that has the smallest time-stamp
• **Event methods:** for each even type, a method to update system state when that event occurs
• **Random-variate generators** methods to generate samples from desired probability distributions
• **Main program :**to maintain overall control of the event –scheduling algorithm
• **Report generator:** a method that computes summary statistics from cumulative statistics and prints a report at the end of the simulation

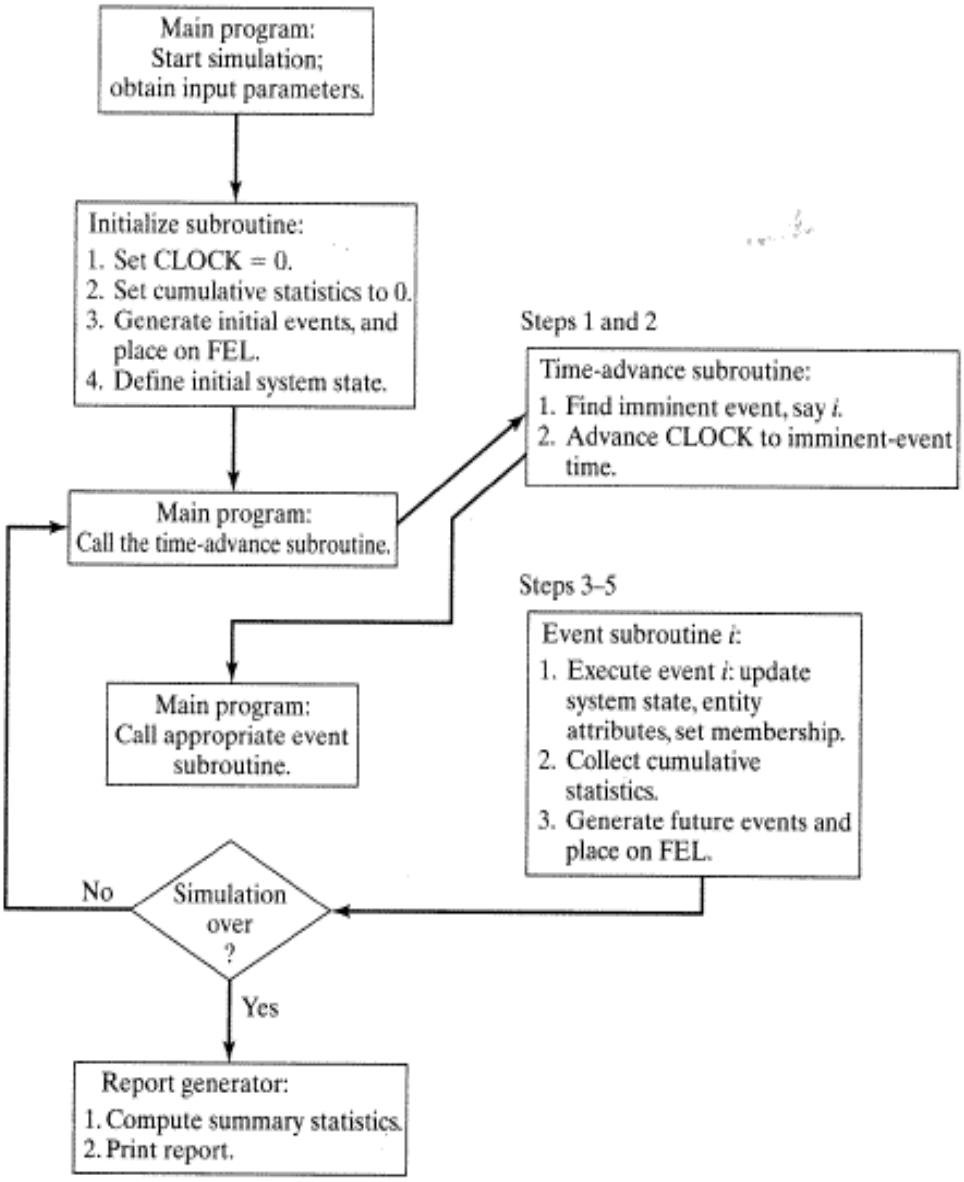

Figure 4.1 Overall structure of an event-scheduling simulation program.

Simulation in GPSS: GPSS, is a discrete time simulation general-purpose programming language, where a simulation clock advances in discrete steps. A system is modelled as transactions enter the system and are passed from one service (represented by blocks) to another. It is used primarily as a process flow oriented simulation language; this is particularly well-suited for problems such as a factory.

GPSS was developed by IBM's Geoffrey Gordon at the beginning of the 1960s.

He named it **G**ordon's **P**rogrammable **S**imulation **S**ystem. The name was changed when IBM decided to release it as a product.[2]

The "General Program" part of the new name was to create a standard in waiting-line simulations.

The original releases were for IBM's 7044 & 7090 mainframes.

Some Blocks of GPSS:

- GENERATE
- TERMINATE
- SEIZE
- RELEASE
- QUEUE
- DEPART
- ADVANCE
- ASSIGN
- START

GENERATE BLOCK:

This block will produce a flow of transactions with inter-arrival times determined by the attribute values. The label is optional. The distribution of inter-arrival times follows a uniform probability distribution.

SYNTAX:

- *line number label* GENERATE A,B,C,D,E

ATTRIBUTES:

- A = average value of uniform distribution
- B = half-width of uniform distribution
- C = time delay before first transaction is generated
- D = maximum number of transactions generated
- E = priority allocated to transactions

QUEUE BLOCK:

This block will instruct GPSS to start gathering queuing statistics on the queue named in its attribute value.

The label is optional but may be necessary if you have to refer to this line from somewhere else in the program

SYNTAX:

- *line number label* QUEUE A

ATTRIBUTES:

- A = name of queue (for example: garage)

If a transaction arriving at the queue block cannot proceed because it is blocked by the next stage, then it will stay in the queue block until it can gain entry to the next stage.

DEPART BLOCK:

This block instructs GPSS that a transaction is leaving the queue named in it's attribute value. This is necessary in order to compile the statistics on the queue. The label is optional.

SYNTAX:

- *line number label* DEPART A

ATTRIBUTES:

- A = name of the queue (for example: checkout)

SEIZE BLOCK:

This blocks allows the transaction to seize a facility if it is free. Thus it may be a car "seizing" a "facility" such as a petrol pump or a customer in a supermarket "seizing" a "facility" such as the checkout assistant. When the car or customer is being serviced by the facility, then it is said to "own the facility". The label is optional.

SYNTAX:

- *line number label* SIZE A

ATTRIBUTES:

- A = name of facility (for example: pump)

** A transaction can only seize a facility if it is free or else wait until the owning transaction releases it.*

RELEASE BLOCK:

A transaction entering this block informs GPSS that it is giving up ownership of the facility named in its attribute value. The label is optional.

SYNTAX:

- *line number label* RELEASE A

ATTRIBUTES:

- A = name of facility (for example: runaway)

* By giving up ownership of the facility, the transaction makes it available for another transaction that may be waiting to use it.

ENTER BLOCK:

This Block instructs GPSS that a transaction has entered STORAGE. The name of storage is given by the first attribute value. The second attribute value gives the amount the storage will be incremented by, when the transaction enters the ENTER block. A STORAGE must be declared at the beginning of a program. For example:

100 Warehouse STORAGE 25

In the 'label' section you must give the STORAGE a name so that the ENTER block can refer to it. The "verb" is STORAGE and the attribute value A, which is 25 in this example, states the maximum capacity of the Warehouse.

SYNTAX:

- *line number label* ENTER A,B

ATTRIBUTES:

- A = name of the storage (for example: warehouse)
- B = increment storage by this value

LEAVE BLOCK:

This block instructs GPSS that a transaction is leaving a STORAGE. The first attribute gives the name of the STORAGE and the second attribute decrements the storage by the value of the attribute.

SYNTAX:

- *line number label* LEAVE A,B

ATTRIBUTES:

- A = name of the storage (for example: warehouse)
- B = Decrement storage by the value

ADVANCE BLOCK:

This block represents the servicing of a transaction. The servicing times follow a uniform probability distribution. The label is optional.

SYNTAX:

- *line number label* ADVANCE A,B

ATTRIBUTES:

- A = average value of uniform distribution
- B = half-width of uniform distribution

* A transaction entering this block will be delayed by a time interval chosen at random from the specified probability distribution.

TERMINATE BLOCK:

This block destroys any transaction entering it and removes it from computer memory. Each time a transaction enters this block it decrements a counter by an amount equal to its attribute value. The counter is set by the user upon starting the simulation.

SYNTAX:

- *line number label* TERMINATE A

ATTRIBUTES:

- A = decrements simulation counter by this amount

* When the counter, set at the beginning of the simulation, reaches zero then the simulation is complete and a statistical report is produced on the outcome of the simulation

TEST BLOCK:

This block can test the logical condition of a queue or storage according to a particular reference value. If a transaction enters the TEST block, the block will check this condition and if it is true, it will send the transaction to one destination in the program and if the condition is false, it will send it to another. For example:

158 TEST LE S$ONHAND, 20, OK

In this example, the STORAGE names ONHAND is checked to see if the amount in storage is less than or equal to (LE) 20. If it is, then the transaction proceeds to the next block in the program. If the condition is false, then the transaction is sent to the program line which has the label "OK" attached to it. The LE part of the verb can be replaced by GE, G, L, E and NE with the usual logical meaning (for example: greater than or equal to, greater than, less than, equal to and not equal to).

SYNTAX:

- *line number label* TEST O A,B,C

ATTRIBUTES:

- A = value and name of the block being referenced
- B = reference value
- C = destination for the transaction if the logical condition is not satisfied

TRANSFER BLOCK:

This block will take transactions entering it and transfer them to each of two different destinations according to laid down proportions. For example:

200 TRANSFER 0.95, EXIT, REPAIR

In this case 95% of all transactions entering the TRANSFER block will go to the program line labelled REPAIR and 5% will go to the program line labelled EXIT. If the second attribute "EXIT" is replaced by a "comma", then the 5% will go to the next block in the program.

SYNTAX:

- *line number label* TRANSFER A,B,C

ATTRIBUTES:

- A = probability value (0 to 1)
- B = proportion of (1-A) transactions transferred to this labelled location
- C = proportion A transactions transferred to this labelled location

---

Scalable Simulation Framework (SSF)

SSF provides a single, unified interface for discrete-event simulation (the SSF API). Object-oriented models that utilize and extend the framework can be portable across SSF-compliant simulation environments. This maximizes the potential for direct reuse of model code, while minimizing dependencies on a particular simulator kernel implementation. In addition to its concrete modeling applications, the API also functions as an abstract target for compilation of models specified in higher-level modeling languages or graphical modeling environments.

The framework's primary design goal was to support high performance simulation. SSF makes it possible to build models that are efficient and predictable in their use of space, able to transparently utilize parallel processor resources, and scalable to very large collections of simulated entities.

Experimentation and Statistical-Analysis Tools

Data for statistical studies are obtained by conducting either experiments or surveys. Experimental design is the branch of statistics that deals with the design and analysis of experiments. The methods of experimental design are widely used in the fields of agriculture, medicine, biology, marketing research, and industrial production.

Statistics are mathematical computations used to analyze data. Tools of statistical analysis can describe, summarize and compare data. There are various tools that can analyze statistical data. These range from relatively simple computations to advanced analysis. Basic analyses can be easily computed, while more advanced methods require a solid understanding of advanced statistics as well as specialized computer software.

Descriptive Analysis

Descriptive analysis uses specific tools to describe data. These are relatively simple calculations that give a basic picture of what the data looks like overall. Descriptive tools include: frequency, percentages and measures of central tendency. Frequency tells how many times something has occurred in a data set. Percentages are calculations that show a proportion. Measures of central tendency are represented by the mean, median and mode. These tools describe the central point (median), the most common (mode) or the average (mean) for a specific variable.

Moderate Analysis

Moderate statistical analysis tools look at the relationships between variables -- what the nature of these relationships are and if they are significant. These include correlation and regression. A correlation describes the relationship between two variables as well as the direction and strength of that relationship. Regression can show if a variable predicts another variable. Like correlation, however, regression does not show causation.

STATISTICAL ANALYSIS FEATURES

- **DESCRIPTIVE** statistics (mean, variance, standard deviation, etc.).
- **FREQUENCY** analysis including frequencies table, descriptive statistics, percentiles table, barchart, pie chart, Pareto chart, histogram, normal probability plot, box-&-whiskers plot and cumulative distribution plot.
- **CROSSTABULATION**: normal crosstabulation and **INTER-RATERS AGREEMENT** table,nominal statistics (chi-square, Pearson's Phi, Goodman-Kruskal's Gamma, Contingency coefficient), ordinal statistics (Kendall's tau-b and tau-c, Pearson's R, symmetric and asymmetric Somers' D, Dxy and Dyx), inter-raters agreement statistics (percentage of agreement, Cohen's Kappa, Scott's Pi, Krippendorf's r and R-bar, free marginal correction for nominal and ordinal measure), 3-D bar chart.
- **BREAKDOWN** analysis with multiple Box-and-Whiskers plot.
- **MULTIPLE RESPONSES** frequency and crosstabulation analysis.
- **PAIRED AND INDEPENDENT T-TESTS** with effect size measures (r and d), error bar graph, barchart, dual histogram.
- **ONEWAY ANALYSIS OF VARIANCE** with post hoc tests (LSD, Tukey's HSD, Scheffé's test), effect size measures, error bar graph, barchart, deviation chart.
- **GLM ANOVA/ANCOVA** (up to 5 factors and covariates) including detailed ANOVA table, 3 different adjustment methods for unequal cell sizes (regression, nonexperimental, hierarchical), multiple regression statistics, test of change of R-Square, regression equation (B, standard error of B, beta, confidence, interval of B, zero-order, semi-partial and partial correlations, tolerance level, F, significance), residuals caseplot with Durbin-Watson statistic, standardized residuals scatterplot, normal probability plot of residuals, ability to save predicted and residual values.
- **CORRELATION MATRIX** including covariance and cross product deviation, user-specified confidence interval, scatterplot matrix.
- **PARTIAL CORRELATION MATRIX** with interactive correlation matrix for inclusion of exclusion of control variables, computation of confidence intervals, etc.
- **REGRESSION** analysis including linear and 7 nonlinear regressions (quadratic, cubic, 4th and 5th degree polynomial, logarithmic, exponential, inverse), regression equation, analysis of variance,

Durbin-Watson statistics, scatterplot, residuals caseplot, standardized residuals scatterplot, normal probability plot of residuals, ability to save predicted and residual values.

- **MULTIPLE REGRESSION** analysis including 5 different regression methods (hierarchical entry, forward selection, backward elimination, stepwise selection, enter all variables), P to enter, P to remove, and tolerance criteria, ANOVA table, test of change ANOVA table, regression equation (B, standard error of B, beta, confidence, interval of B, zero-order, semi-partial and partial correlations, tolerance level, F, significance), residuals caseplot, Durbin-Watson statistic, standardized residuals scatterplot, normal probability plot of residuals, ability to save predicted and residual values.
- **TIME SERIES** analysis including data transformation (ex.: remove mean, lag, etc.), auto-correlation diagnostic (ACF and PACF plot), smoothing techniques (moving average,running median), control bars with user-specified confidence interval.
- **SINGLE-CASE EXPERIMENTAL DESIGN** analysis with descriptive statistics, interrupted time-series graph, various graphical judgmental aids such as smoothing (moving average and running median), trend lines and control bars.
- **RELIABILITY** analysis with item, inter-item and item-total statistics, split-half reliability statistics, internal consistency measures (Cronbach's alpha, etc.).
- **CLASSICAL ITEM ANALYSIS** for multiple-choice item questionnaires.
- **FACTOR ANALYSIS** including principal components analysis and image covariance factor analysis, Q-type factor analysis, varimax rotation, scree plot, etc..
- **SENSITIVITY ANALYSIS** with false-positives and false negatives statistics, sensitivity and specificity statistics, ROC curve (Receiver operating characteristics), error rate graph.
- **NONPARAMETRIC** analysis including binomial test, one sample chi-square test, runs test, McNemar test, Mann-Whitney U test, Wilcoxon t-test, sign test, Kruskal-Wallis ANOVA, Friedman two way ANOVA, Kolmogorov-Smirnov test for 2 samples and goodness of fit test, Moses test of extreme reactions, median test (2 or more samples).
- **NONPARAMETRIC ASSOCIATION MATRIX** including Spearman's R, Somer's D, Dxy and Dyx, Goodman Kruskal's Gamma, Kendall's Tau-a, Tau-b, Kendall Stuart's Tau-c, etc.
- **BOOTSTRAP RESAMPLING** analysis including resampling of 7 univariate and 21 bivariate estimators, descriptive statistics, percentile table, nonparametric confidence intervals, nonparametric power analysis, variable sample size, random sampling simulation, histogram.
- **FULL ANALYSIS BOOTSTRAP** resampling on almost every analysis (frequency, crosstab, multiple regression, reliability, nonparametric tests, etc.).

OTHER FEATURES

- Integer weighting of cases using another variable.
- Runs LOGISTIC, a freeware logistic regression program written by Gerard L. Dallal.
- SIMCALC probability calculator computes probabilities for 9 types of test/distribution as well as confidence intervals for proportions, mean, and correlation.

DATA MANAGEMENT FEATURES

The data window is a spreadsheet like data editor where values can be entered, browsed, or edited.

- Data file can store up to 2035 variables (or fields).
- Supports plain text as well as Rich Text Format documents.

- Imports comma or tab separated text files, DBase, FoxPro, Excel, MS Access, Paradox, Lotus, Quattro Pro, SPSS/PC+, and SPSS for Windows files.
- Exports comma or tab separated text files, DBase, FoxPro, Excel, MS Access, Paradox, Lotus, Quattro Pro, SPSS/PC+, and XML files.
- Allows merging and aggregation of data files.
- Supports variable and value labels and up to 3 missing values.
- Cases can be filtered using complex xBase expressions.
- Data grid may be sorted on one or several variables.
- Customizable grid provides alternate view of the data file
- Supports data transformation (including conditional transformation), recoding, ranking. Provides more than 50 transformation functions including trigonometric, statistical, random number functions.

OUTPUT MANAGEMENT FEATURES

The Notebook window displays the statistical output for all analysis performed during a session. The notebook metaphor provides an efficient way to browse and manage outputs.

- The text output of each analysis is displayed on a separate page.
- Each page can be annotated or edited.
- Empty pages can be inserted to put down ideas or remarks, sketch an analysis plan, or write down interpretation of results.
- Tabs can be added to create sections in the notebook allowing storage of different kinds of analysis in different sections of the notebook.
- An index of all analysis is automatically generated. This index can be used to quickly locate and go to a specific page, move pages within the notebook, or delete some pages.
- Rich Text notebook allows one to change font attributes (bold, underline, strikeout, italic) and font colors.
- Highlight tool allows to color passages of text.
- Notebooks can be exported in Rich Text Format or in plain text.

CHART CREATION AND MANAGEMENT FEATURES

All high-resolution charts created during a session are displayed in the Chart window. This window can be used to view the charts and perform various operations on individual charts or on the entire collection of charts. For example, you can modify the various chart attributes, save those charts to disk, export them to another application using the clipboard or disk files, or print them. It is also possible to delete a specific chart or to modify the order of those charts in this window.

- Control of axis, titles, legends, colors, lines, etc.
- Charts can be imported on disk or copied to the clipboard in Window bitmap or metafile format
- Charts can also be copied as tab separated values and imported by another charting application.

SCRIPT, EDUCATIONAL, & MULTIMEDIA FEATURES

The script window is used to enter and edit commands. Those commands can be either read from a script file on disk, typed in by the user or automatically generated by the program. When used with the RECORD feature, the script can also be used as a log window to keep track of the analysis performed during a session. Those commands may then be executed again, providing an efficient way to automate statistical analysis. Additional commands also allows one to create demonstration programs, computer assisted teaching lessons, and even computer assisted data entry.

- Statistical analysis, data filtering and transformation commands
- Record script feature to automatically generate commands corresponding to operations performed with menus and dialog boxes.
- Flow control features such as IF-THEN-ELSE statements, GOTO or GOSUB commands, RUN command to run external programs, etc.
- Can read, write, and perform mathematical operations on user defined variables or any database field
- Create menus, text boxes, input boxes, dialog boxes, multiple items questions, etc. (responses from a user can be stored in memory variables or in a data file)
- Multimedia features: play sound (.WAV), music (.MDI), and movie (.AVI) files, display graphics (.BMP) and text.