

STATWIN: AN OBJECT-ORIENTED APPROACH TO
ESTIMATE THE TOXICITY OF EFFLUENTS ON
FRESHWATER ORGANISMS

By

JINYU HUANG

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1992

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1995

STATWIN: AN OBJECT-ORIENTED APPROACH TO
ESTIMATE THE TOXICITY OF EFFLUENTS ON
FRESHWATER ORGANISMS

Thesis Approved:

H. Lu

Thesis Adviser

J. Chandler

Mitchell Hein

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

My primary thanks go to Dr. Huizhu Lu, my advisor, for her help and encouragement. I would also like to thank Drs. John P. Chandler and Mitch Neilsen for being my committee members. I also thank Drs. John P. Chandler and Blayne E. Mayfield for providing a teaching assistantship at the Department of Computer Science.

A special thank you goes to Mrs. Marie Hesser, who helped me to correct grammatical errors of my thesis.

And last, but not least, I would like to thank my family. Although my parents Professor Tianxi Huang, Professor Huiying Liu and my brother Wei Huang are in China, hundreds and thousands of miles away from me, their love and encouragement have been supporting me throughout my studies in this country. Special thanks go to my husband Shuhong Chen; without his love, support and encouragement, I would have never made it.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
Motivation	1
Objectives	2
About This Thesis	2
II. LITERATURE REVIEW	4
Statistical Analysis.....	4
The Shapiro-Wilk Test	6
Bartlett’s Test	7
Dunnnett’s Test	8
Bonferroni’s T Test	9
Steel’s Many-one Rank Test	10
Wilcoxon Rank Sum Test	10
Fisher’s Exact Test	11
Linear Interpolation	12
Transformation of the Input Data	13
Arc Sine Square Root Transformation	14
log ₁₀ Transformation	15
Minimum Significant Difference (MSD).....	15
Object-oriented Program Review	16
Objects and Classes	16
Inheritance	17
Overview of Past Work	18
III. DESIGN OF THE STATWIN PROGRAM	27
The Internal Design Overview	27
Internal Structure for Statistical Test	27
Statistical Tables Used in STATWIN.....	28
Transformation of the Data	29
The Statistical Tests	30
The C++ Classes Implemented.....	30
Communication between ST and GUI.....	35

Graphical User Interface (GUI).....	35
The Document-view Architecture	36
The C++ Classes Implemented.....	36
IV. CONCLUSION	44
Summary of STATWIN and Major Improvements.....	44
Possible Extension of STATWIN.....	45
BIBLIOGRAPHY.....	48
APPENDIX	49

LIST OF TABLES

Table	Page
I. Generalized ANOVA Table	8
II. Format for Contingency Table	11
III. Statistical Tables and Tests	29
IV. Transformations and Statistical Tests	30
V. A Summary of Testing Data in the STATWIN Program	51

LIST OF FIGURES

Figure	Page
1. Flow Chart for Statistical Analysis of Fathead Minnow Larval Survival Data.....	19
2. Flow Chart for Statistical Analysis of Fathead Minnow Larval Growth Data.....	20
3. Flow Chart for Statistical Analysis of Fathead Minnow Embryo-larval Data.....	21
4. Flow Chart for Statistical Analysis of <u>Ceriodaphnia</u> Survival Data	22
5. Flow Chart for Statistical Analysis of <u>Ceriodaphnia</u> Reproduction Data	23
6. Flow Chart for Statistical Analysis of Algal Growth Response Data	24
7. The Procedural Paradigm	25
8. The Object-oriented Paradigm	25
9. Inheritance	26
10. Internal Structure of STATWIN Program	28
11. Class Hierarchy for Statistical Tests (ST)	31
12. Document and View	37
13. Internal Structure of GUI for STATWIN	37
14. Main Frame Window of STATWIN	39
15. Splitter Window of STATWIN	40

16. Data Entry of STATWIN	41
17. Plot of STATWIN	42
18. Administration Information Dialog Box of STATWIN	43
19. Start Up Window of STATWIN	53
20. Available Selections of Test Type.....	55
21. Available Selections of Analysis Method.....	56
22. Example Input for <u>Ceriodaphnia</u> Survival Test	57
23. Administration Information Dialog Box	57
24. Tool Bar and Status Bar	59
25. File Save As Dialog Box	59
26. Execution Message Box	60
27. Example of STATWIN Report	62
28. Example of Line Plot	63
29. Example of 2D Bar Plot	63
30. Example of 3D Bar Plot	64
31. About STATWIN Dialog Box	65
32. File Open Dialog Box	65
33. Frame Window without Toolbar and Statusbar	67

CHAPTER I

INTRODUCTION

In the manual of “Short-term Methods for Estimating the Chronic Toxicity of Effluents and Receiving Waters to Freshwater Organisms” [Weber et al. 1989], the U. S. Environmental Protection Agency has developed standard methods for estimating the toxicity of pollutants to aquatic organisms, which are used by the U. S. Environmental Protection Agency regional and state programs. The manual provides information on how to conduct tests biologically and how to analyze data statistically. The author attempts by the object-oriented method to develop a window version program STATWIN to help analyze data automatically by using Microsoft Visual C++ 1.0 [Microsoft 1993]. In the rest of the thesis, the manual also refers to the EPA manual or the EPA document.

Motivation

According to the EPA document, Figure 1 is the flowchart for statistical analysis of test data. Basically, the seven statistical tests involved in the flowchart can be seen as seven different blocks or objects from the object-oriented programming standpoint, which is not primarily concerned with the details of program operation, but deals with the overall organization of the program. Each of the tests, taking care of its own operation and connecting to each other based on the flowchart, would reduce the communication overhead and help maintain the integrity of the program. Also, the author wants to provide a user-friendly environment to the users in terms of pull-down menus, dialog

boxes and drop lists, etc. Therefore, Microsoft Visual C++ 1.0 [Microsoft 1993] is chosen for the purpose of both object-oriented programming and window environment.

Objectives

The objective of STATWIN is to implement a window version program which provides a user-friendly Graphical User Interface (GUI) environment so that users even with little knowledge of statistics can obtain the statistical results by simply inputting the data from biological tests.

About This Thesis

This thesis includes introduction to the general information of each chapter and literature reviews of past works, details the internal design of both statistical tests and graphical user interface of the program, STATWIN. The following chapters are included:

- ◆ Chapter I , the current chapter, describes a brief introduction to the thesis. Author hopes this will lead to a better understanding of the overall organization of the thesis and the program STATWIN.
- ◆ Chapter II presents the information required to the program, including statistical methods, flow chart of an analysis and biological terminology in the EPA document. It also reviews some past work which specifically mentions work of K. L. Smith who wrote TOXY in C in 1991 [Smith, 1991] for the same document, as well as the basic concepts of C++ language, which is used in STATWIN.

- ◆ Chapter III provides the internal structure of STATWIN in detail and explains why STATWIN is split into two parts, which are statistical tests (ST) and graphical user interface (GUI), and how these two components are communicated with each other.
- ◆ Chapter IV summarizes the major improvements of STATWIN and points out the possible extension.
- ◆ Appendix contains information on the data files attached to STATWIN. Additionally, it demonstrates the use of various features of this program.

CHAPTER II

LITERATURE REVIEW

The thesis is based primarily on the EPA manual "Short-term Methods for Estimating the Chronic Toxicity of Effluents and Receiving Waters to Freshwater Organisms" [Weber et al. 1989] Second Edition, which provides the biological and statistical methods for testing toxicity on aquatic organisms.

Statistical Analysis

Hypothesis testing and point estimates are employed for analyzing the toxicity on water organisms. To determine toxicity limits in permits, the following concepts are explained.

No-observed-effect-concentration (NOEC) is the highest concentration which would cause no observable adverse effect on the test organisms. Statistically, the effect is not different from the control.

Lowest-observed-effect-concentration (LOEC) is the lowest concentration that would cause adverse effect on the test organisms. In other words, the effect is statistically different from the control.

The purpose of the hypothesis testing is to get the interval of concentrations, (NOEC, LOEC).

Effective concentration (EC) is a point estimate of concentration which would cause an adverse effect in a given percent of test population. If the adverse effect is death,

the EC becomes Lethal concentration (LC). For example, LC50 is the point estimate that would cause death in 50% of the organisms.

Inhibition concentration (IC) is a point estimate that would cause a given percent reduction in fecundity or growth. For instance, IC50 is the point estimate that would cause 50% reduction in mean young per female or growth.

The EPA manual suggests conducting six tests on water samples to estimate toxicity. They are the following:

1. Fathead minnow larval survival test
2. Fathead minnow larval growth test
3. Fathead minnow embryo-larval test
4. Ceriodaphnia survival test
5. Ceriodaphnia reproduction test
6. Algal growth response test

The six flow charts corresponding to the six tests are in Figure 1 to Figure 6.

Hypothesis testing can be divided into two categories, parametric and nonparametric. T-test with Bonferroni adjustment and Dunnett's test are parametric tests for NOEC and LOEC when normality of data is met, and for homogeneity of variance across the data, while Steel's many-one rank test and Wilcoxon rank sum test with Bonferroni adjustment are nonparametric tests for NOEC and LOEC. If an equal number of replicates occurs among the concentration levels, T-test with Bonferroni adjustment and Steel's many-one rank test are for parametric and non-parametric options, respectively. Dunnett's test and the Wilcoxon rank sum test with Bonferroni adjustment

are for parametric and non-parametric alternatives, respectively, for unequal number of replicates. Before either parametric or nonparametric tests are conducted, the Shapiro-Wilk test is first performed for normality followed by Bartlett's test for homogeneity of variances. The following paragraphs explain each test in detail.

The Shapiro-Wilk Test

The Shapiro-Wilk test is to test if the data are normally distributed [Weber et al. 1989]. The steps are as follows:

1. Center the observations by subtracting the mean of all observations within a concentration from all observations in that observation.

2. Compute D:

$$D = \sum_{i=1}^n (x_i - \bar{x})^2$$

Where x_i is the centered observation and \bar{x} is the mean of all centered observations, and n is the total number of observations, it can be proved that $\bar{x} = 0$.

3. Order the centered observations in ascending order,

$$x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(n)}.$$

where $x^{(i)}$ is the ith order observation.

4. For the total number of observations n, $k = n/2$, check the table for coefficients

$$a_1, a_2, \dots, a_k$$

5. Calculate W, the test statistic:

$$W = \frac{1}{D} * \left\{ \sum_{i=1}^k a_i * (x^{(n-i+1)} - x^{(i)}) \right\}^2$$

6. Compare W to the critical value in the table of the Shapiro-Wilk test statistic at the significant level p and the number of observations n. The data are normally distributed if $W > \text{critical value}$.

Bartlett's Test

Bartlett's test is to test if the variance of each group is equal to that of control [Weber et al. 1989, Bain and Engelhardt 1987]. The steps are as follows:

1. Calculate the statistic B:

$$B = \frac{[(\sum_{i=1}^p v_i) \ln(\bar{s}^2) - \sum_{i=1}^p (v_i * \ln(s_i^2))]}{C}$$

Where V_i = the degrees of freedom (df) for each concentration and the control.

p = the number of concentration including the control.

\bar{s}^2 = the average of variances.

$$C = 1 + \left[\frac{1}{3(p-1)} \right] * \left[\sum_{i=1}^p \frac{1}{v_i} - \frac{1}{\sum_{i=1}^p v_i} \right]$$

$$\ln = \log_e$$

2. If the variances are equal, B is distributed as chi-square with p-1 df. Hence, if $B < \text{critical value}$ is obtained from a chi-square table with (p-1) df, then the variances are equal.

Dunnett's Test

Dunnett's test is to test each mean of concentration with that of control. This test is based on the assumptions that data are normally distributed and variances are equal. The number of replicates for each concentration is also required to be equal [Weber et al. 1989]. The following shows the steps for Dunnett's test.

1. Generate an ANOVA Table I.

TABLE I
GENERALIZED ANOVA TABLE

Source	Degrees of Freedom	Sum of Square (SS)	Mean Square (MS)
Between	$p - 1$	SSB	$S_B^2 = \frac{SSB}{p-1}$
Within	$N - p$	SSW	$S_W^2 = \frac{SSW}{N-p}$
Total	$N - 1$	SST	

Where p is the number of concentrations including the control.

Total sum of square:
$$SST = \sum_{i,j} y_{ij}^2 - \frac{G^2}{N}$$

Between sum of square:
$$SSB = \sum_i \frac{T_i^2}{n_i} - \frac{G^2}{N}$$

Within sum of square:
$$SSW = SST - SSB$$

Where y_{ij} = the jth observation for the ith concentration.

G = the grand total of all observations, $G = \sum_i T_i$

T_i = the total for concentration i.

n_i = the number of replicates for concentration i.

N = The total number of observations.

2. Calculate statistic t for each concentration and the control.

$$t_i = \frac{\bar{y}_1 - \bar{y}_i}{S_w \sqrt{\frac{1}{n_1} + \frac{1}{n_i}}}$$

Where \bar{y}_1 = control mean.

\bar{y}_i = mean for ith concentration.

S_w = square root of within mean square.

n_1 = number of replicates for the control.

n_i = number of replicates for concentration i.

3. If $t_i >$ critical value that comes from Dunnett's T table with $df = N - p$ and

$(p - 1)$ concentrations, then the mean for concentration i is less than the mean for control.

Bonferroni's T test

The assumptions for Bonferroni's t test are the same as the Dunnett's test except that the number of replicates for each concentration and control are unequal [Weber et al. 1989].

1. The same as step 1 in Dunnett's test.
2. The same as step 2 in Dunnett's test.
3. If $t_i >$ critical value is obtained from table of Bonferroni's T test with $df = N - p$ and $(p - 1)$ concentrations, then the mean of concentration i is less than the control mean.

Steel's Many-one Rank Test

Steel's many-one rank test can be carried out when the assumptions of normality and equal variances are not met and number of replicates for each concentration are the same [Weber et al. 1989]. In other words, Steel's many-one rank test is a nonparametric test. The following gives the detailed steps of the test.

1. Combine the observations for each concentration and control and arrange them in ascending order.
2. Assign the ranks to the ordered observation. For example, 1 is assigned to the smallest, etc. If ties occur, give the average rank to the observations.
3. Calculate the sum of the ranks, S , in each concentration and the control.
4. If $S <$ critical value that is from the table of Steel's many-one rank sum test, then the group is considered to be significantly different from the control.

Wilcoxon Rank Sum Test

The Wilcoxon rank sum test has the same assumptions for normality and variances of the Steel's many-one rank sum test [Weber et al. 1989]. When the number of replicates of concentrations are not equal, the Wilcoxon rank sum test is the choice for

computing (NOEC, LOEC). The detailed steps for the Wilcoxon rank sum test are as follows:

1. The same as step 1 in Steel's many-one rank test.
2. The same as step 2 in Steel's many-one rank test.
3. If $S < \text{critical value}$ that comes from the table of Wilcoxon rank sum test, then the group is significantly different from the control.

Fisher's Exact Test

The Fisher's exact test is to test if the concentration mean is different from the control mean [Weber et al. 1989]. The test follows the steps described below.

1. First establish a 2*2 contingency table along with two conditions for the table II: a)

$$A \geq B, \text{ b) } \frac{a}{A} \geq \frac{b}{B}.$$

TABLE II
FORMAT FOR CONTINGENCY TABLE

	Successes	Failures	Number of observations
Row 1	a	A - a	A
Row 2	b	B - b	B
Total	a + b	(A + B) - (a + b)	A + B

2. First look up a value b in the table with A , B , and a , then compare to the value b given in the 2×2 contingency table. If $b \leq$ critical value from the table, then the concentration is significantly different from the control.

Linear Interpolation

Linear interpolation is for the point estimate of concentration that causes a given percent reduction of the data. For example, IC25 is the point estimate causing 25% reduction in the growth data. It assumes that data are monotonically non-increasing. In cases where the data are not following the non-increasing pattern, the means of observations need to be smoothed by averaging adjacent means [Weber et al. 1989].

Linear interpolation method is described as follows:

1. Let \bar{y}_1 be the control mean. If the mean of observations at the lowest level \bar{y}_2 is less than or equal to the \bar{y}_1 , then \bar{y}_2 is used. If $\bar{y}_2 > \bar{y}_1$, then the average of \bar{y}_1 and \bar{y}_2 is used for both control mean M_1 and the lowest concentration M_2 . The mean from next higher concentration \bar{y}_3 is compared to the M_1 . If $\bar{y}_3 \leq M_1$, then \bar{y}_3 is used. Otherwise, \bar{y}_3 needs to be averaged with M_1 and M_2 so that

$$M_1 = M_2 = M_3 = \frac{M_1 + M_2 + M_3}{3} .$$

2. Continue this method for all the mean responses.

3. A given percent reduction IC $_p$ can be calculated as follows:

$$IC_p = C_J + [M_1 * (1 - p/100) - M_J] * \frac{C_{J+1} - C_J}{M_{J+1} - M_J} .$$

Where M_1 = Smoothed mean response for the control.

M_J = Smoothed mean response for the concentration J.

M_{J+1} = Smoothed mean response for the concentration J+1.

C_J = The tested concentration whose observed mean response is greater than $M_1(1-p/100)$.

C_{J+1} = The tested concentration whose observed mean response is less than $M_1(1-p/100)$.

Transformation of the Input Data

Of all the statistical analysis as follows:

1. Fathead minnow larval survival test,
2. Fathead minnow larval growth test,
3. Fathead minnow embryo-larval test,
4. Ceriodaphnia survival test,
5. Ceriodaphnia reproduction test, and
6. Algal growth response test,

input data of some of the tests need to be transformed when normality or homogeneity of variance or both are not met, so that data can be analyzed by a parametric method such as Dunnett's test if number of replicates are equal, or T-test with Bonferroni adjustment if the number of replicates are not equal, rather than a non-parametric procedure such as Steel's many-one rank test for equal number of replicates or Wilcoxon rank sum test with Bonferroni adjustment for unequal number of replicates of data. Transformations of arc

sine square root and \log_{10} are employed in the STATWIN program based on the suggestion of the EPA document, which is the foundation of the program STATWIN.

Arc Sine Square Root Transformation

Arc sine square root transformation is used for mortality data, such as fathead minnow larval survival analysis and fathead minnow embryo-larval analysis for hypothesis testing, which deal with survival proportion [Weber et al. 1989]. Whenever the proportion mortality is 0 or 1, a special modification must be used for transformation.

To calculate the arc sine square root transformation, the number of animals per treatment should be provided for proportion dead 0 or 1. Proportion mortality is also referred to response proportion (RP), which is the ratio of number of dead organisms to the total number of exposed. The result of arc sine square root transformation, which is an angle, is represented in radians. An explanation of the transformation is provided below.

Example: Suppose if 8 out of 20 animals at a given concentration die, then,

$$\begin{aligned} 1. \quad \text{RP} &= 8/20 \\ &= 0.40. \end{aligned}$$

Since $0 < \text{RP} < 1$,

$$\begin{aligned} \text{Angle} &= \arcsin(0.40)^{0.5} \\ &= \arcsin (0.6325) \\ &= 0.6847 \text{ radians.} \end{aligned}$$

2. If $\text{RP} = 0$, the following formula is for calculating the angle.

Angle (in radians) = $\arcsin (1/4N)^{0.5}$, where N is the number of animals per treatment.

if N = 20, then,

$$\begin{aligned}\text{Angle} &= \arcsin (1/4*20)^{0.5} \\ &= \arcsin(0.118) \\ &= 0.1120 \text{ radians.}\end{aligned}$$

3. If RP = 1.0, then,

$$\text{Angle} = 1.5708 \text{ radians} - (\text{radians for RP} = 0)$$

Using above value:

$$\begin{aligned}\text{Angle} &= 1.5708 - 0.1120 \\ &= 1.4588 \text{ radians.}\end{aligned}$$

\log_{10} Transformation

\log_{10} transformation is performed on data for algal growth analysis.

Minimum Significant Difference (MSD)

The minimum significant difference (MSD) is defined as follows in order to quantify the sensitivity of the test [Weber et al. 1989].

$$\text{MSD} = d S_w (1/N_1 + 1/N)^{0.5}$$

where d = the critical value for the Dunnett's test,

S_w = the square root of the within mean square,

N_1 = the number of replicates for the control,

and N = the number of replicates for each concentration, assuming an equal number of replicates at each concentration.

In the case where input data have been transformed, MSD in untransformed units can be calculated as the following example illustrates:

Example: Assume that arc sine square root transformation is used in the test, and MSD in transformed unit named $MSD_t = 0.087$. If the transformed control mean $ControlMean_t = 0.714$, then

1. the control mean in untransformed units

$$ControlMean_u = [\sin (0.714)]^2, \text{ and}$$

2. the untransformed difference

$$\begin{aligned} D_u &= [\sin (0.714 - 0.087)]^2 \\ &= [\sin (0.627)]^2 \\ &= 0.344 \end{aligned}$$

3. Finally, the untransformed MSD, represented by MSD_u is determined by the formula provided below:

$$\begin{aligned} MSD_u &= ControlMean_u - D_u \\ &= 0.429 - 0.344 \\ &= 0.085. \end{aligned}$$

Object-oriented Program Review

Objects and Classes

A program is about data and operations. For a traditional procedural language like C, given global data, they are accessible to all functions as shown in Figure 7 [Lafore 1991]. Data might be changed or corrupted by functions that should not change it, as a result of the procedural paradigm. Another problem is if the way the data is stored needs to change; if so, all the functions that access them also need to change. Consequently, these make the procedural program hard to write, organize, debug and maintain.

Compared to the procedural languages, object-oriented languages are much easier to organize and maintain. The fundamental idea of the object-oriented languages is to combine the data and functions into a single unit called an object or a block. The relationship of objects is shown in Figure 8 [Lafore 1991]. Notice in Figure 8 that communication between each object and also overall organization are much simpler and data is accessed accurately.

In an object-oriented language like C++, an object is deemed to be a member of a class which is a collection of functions and data on which those functions operate. Importantly, class is one of the key concepts in C++ language.

Inheritance

Sometimes, several classes share something in common and each class has its own characteristics as well. This idea leads to another important concept in object-oriented program (OOP) -- inheritance. In C++, inheritance is implemented through base class and its derived classes. The idea of inheritance is shown in Figure 9 [Lafore 1991]. Note that

base class has features A and B, which are common to all derived classes, but each derived class also has its own features.

Overview of Past Work

The EPA methods of measuring the toxicity have been implemented in computer programs to automate the series of statistical analysis. The earliest versions were written in IBM PC basic by D. L. Weiner [Weber et al. 1987] and in FORTRAN for IBM compatible PCs [Weber et al. 1989].

Keith L. Smith wrote a statistical program TOXY for his Master of Science degree in 1991 using C [Smith, 1991], which is a traditional structured programming language, which emphasizes doing things such as reading data from the keyboard, and checking for errors, etc., instead of focusing on the overall organization of the program. TOXY by Smith provides a graphical user interface (GUI) which includes pull-down menus, dialog boxes and messages for data entry. End users have six different statistical analyses to choose from based on the EPA suggestions. In addition, Smith mentioned graphical plots in his thesis with no implementation in the TOXY. To run TOXY, a special set up program needs to run first so the executable program and the initial working directory, color and font are in the same directory.

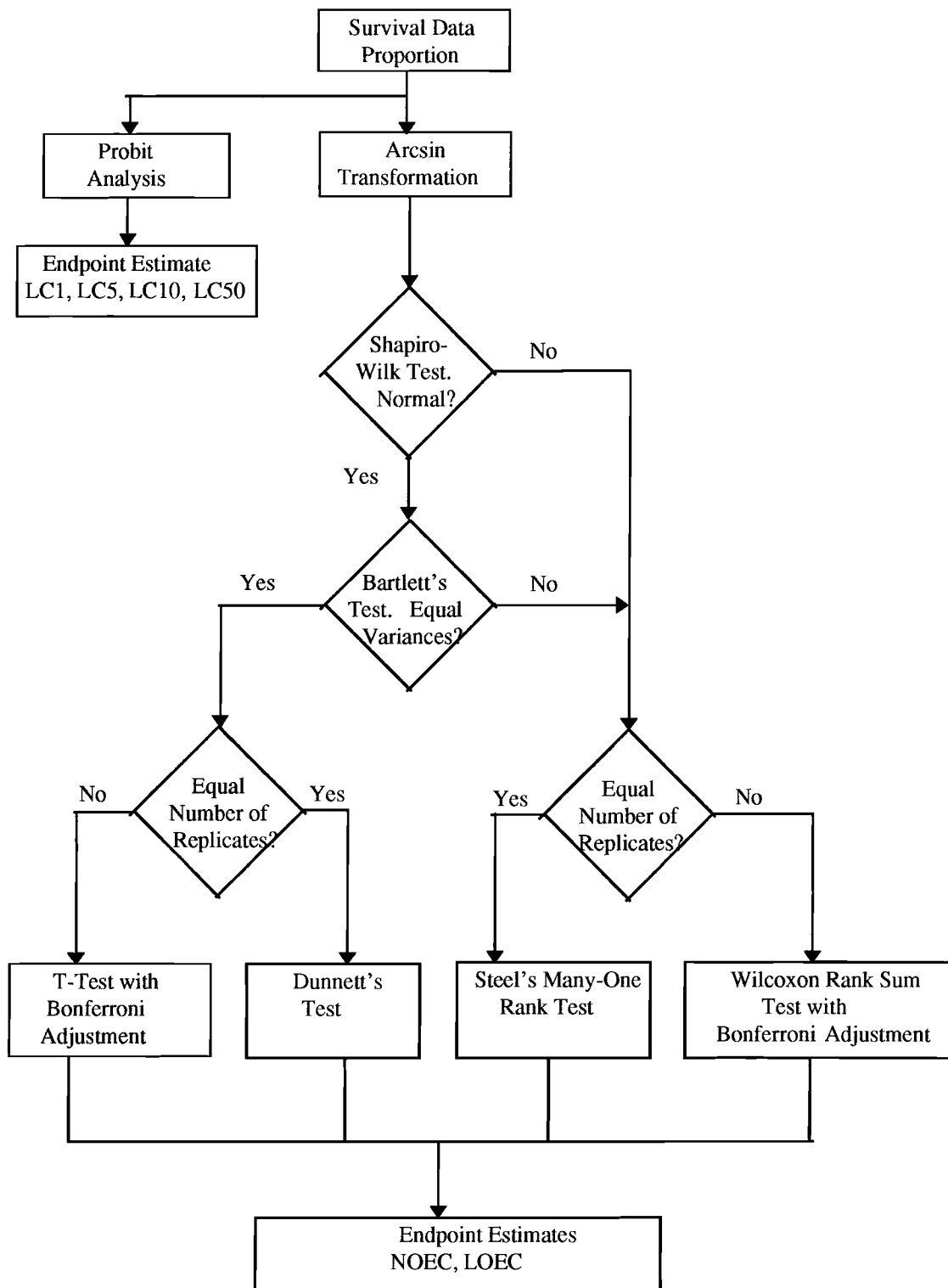


Figure 1. Flow Chart for Statistical Analysis of Fathead Minnow Larval Survival Data

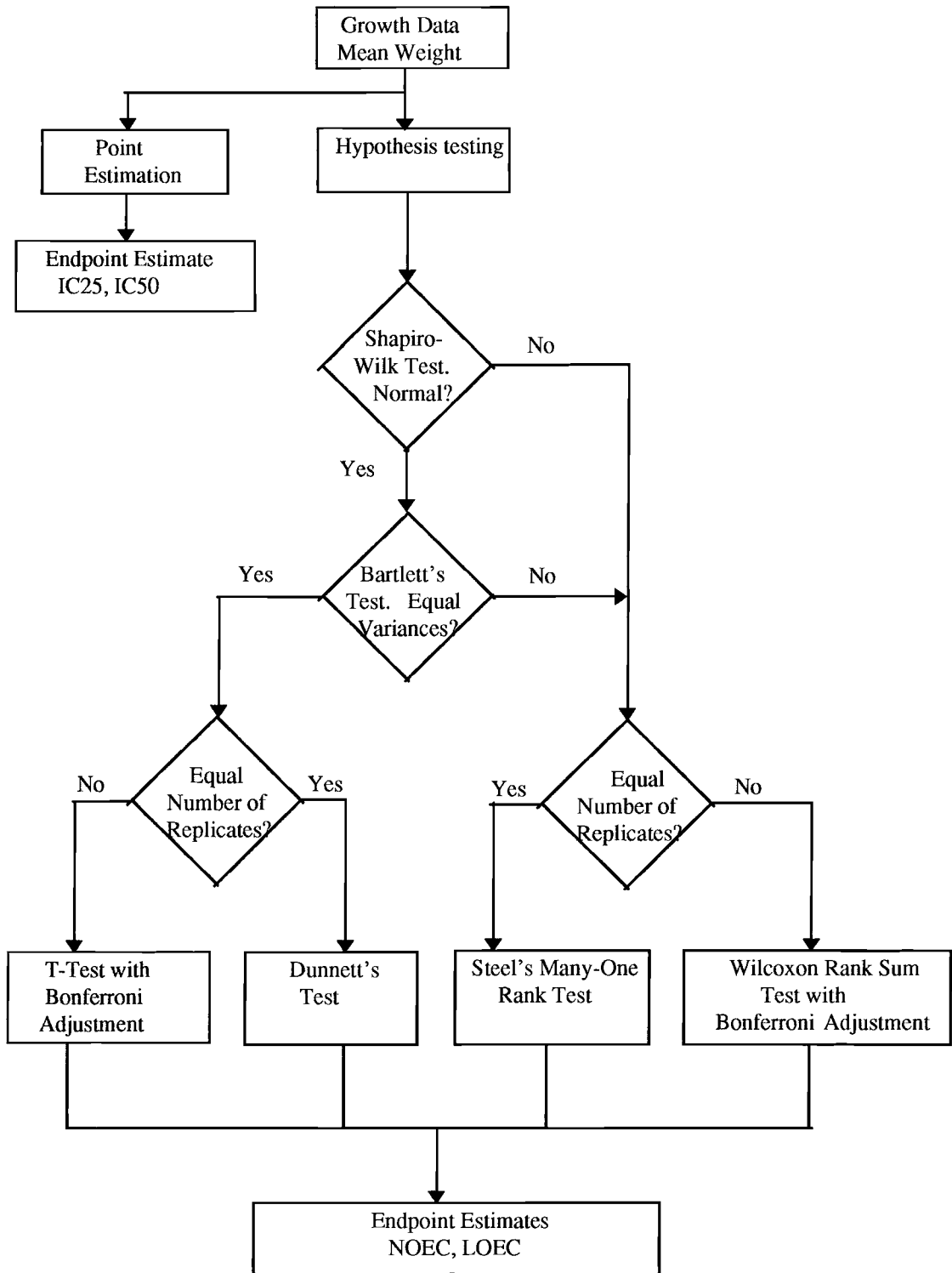


Figure 2. Flow Chart for Statistical Analysis of Fathead Minnow Larval Growth Data

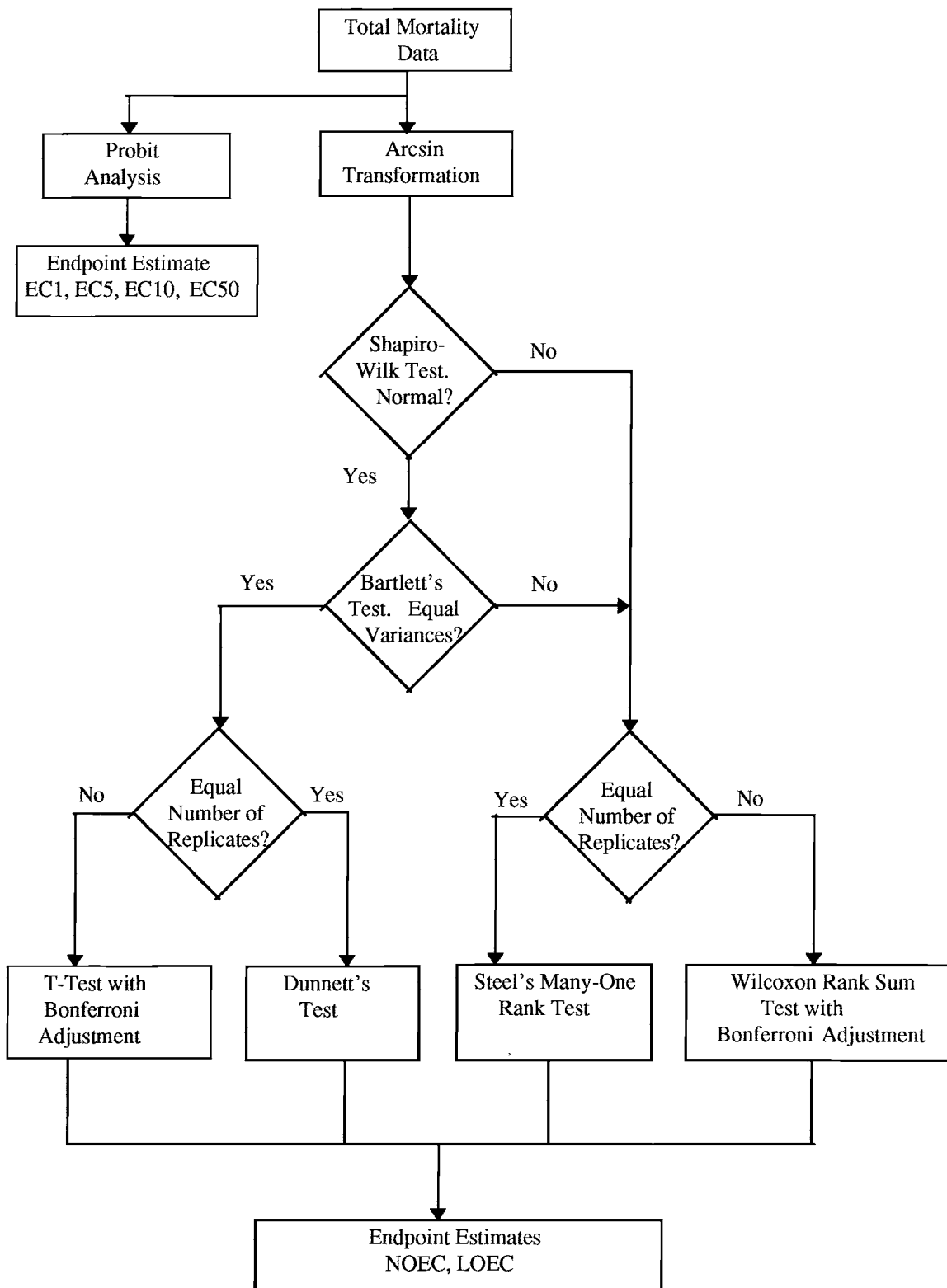


Figure 3. Flow Chart for Statistical Analysis of Fathead Minnow Embryo-larval Data

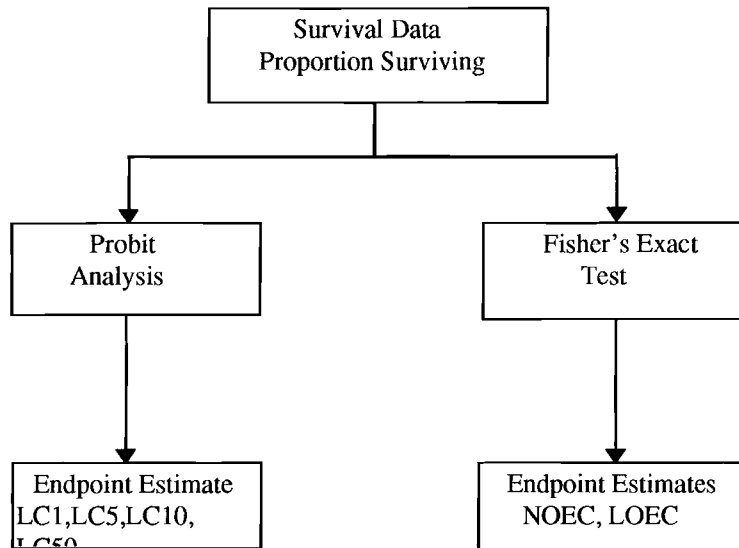


Figure 4. Flow Chart for Statistical Analysis of Ceriodaphnia Survival Data.

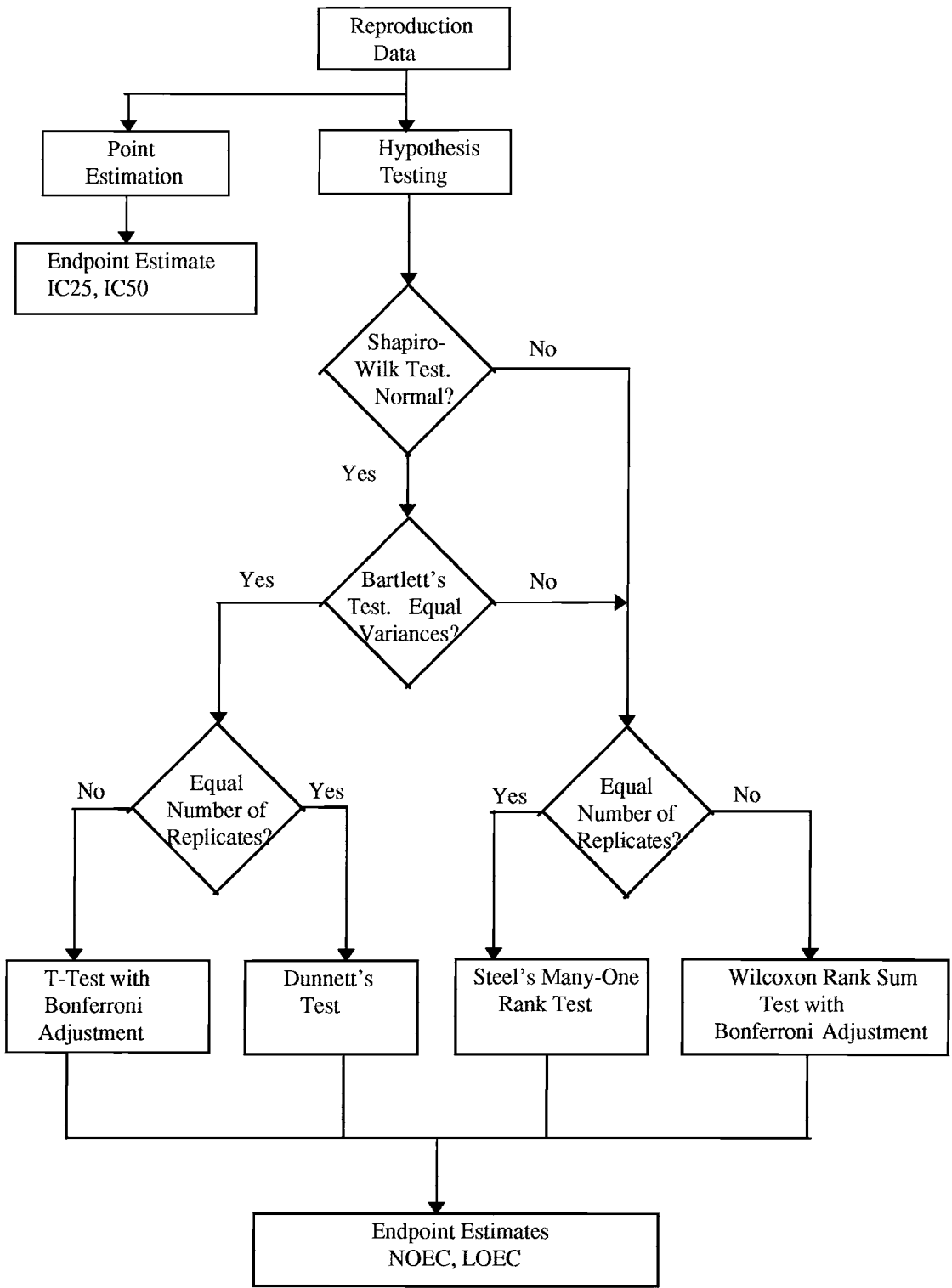


Figure 5. Flow Chart for Statistical Analysis of *Ceriodaphnia* Reproduction Data

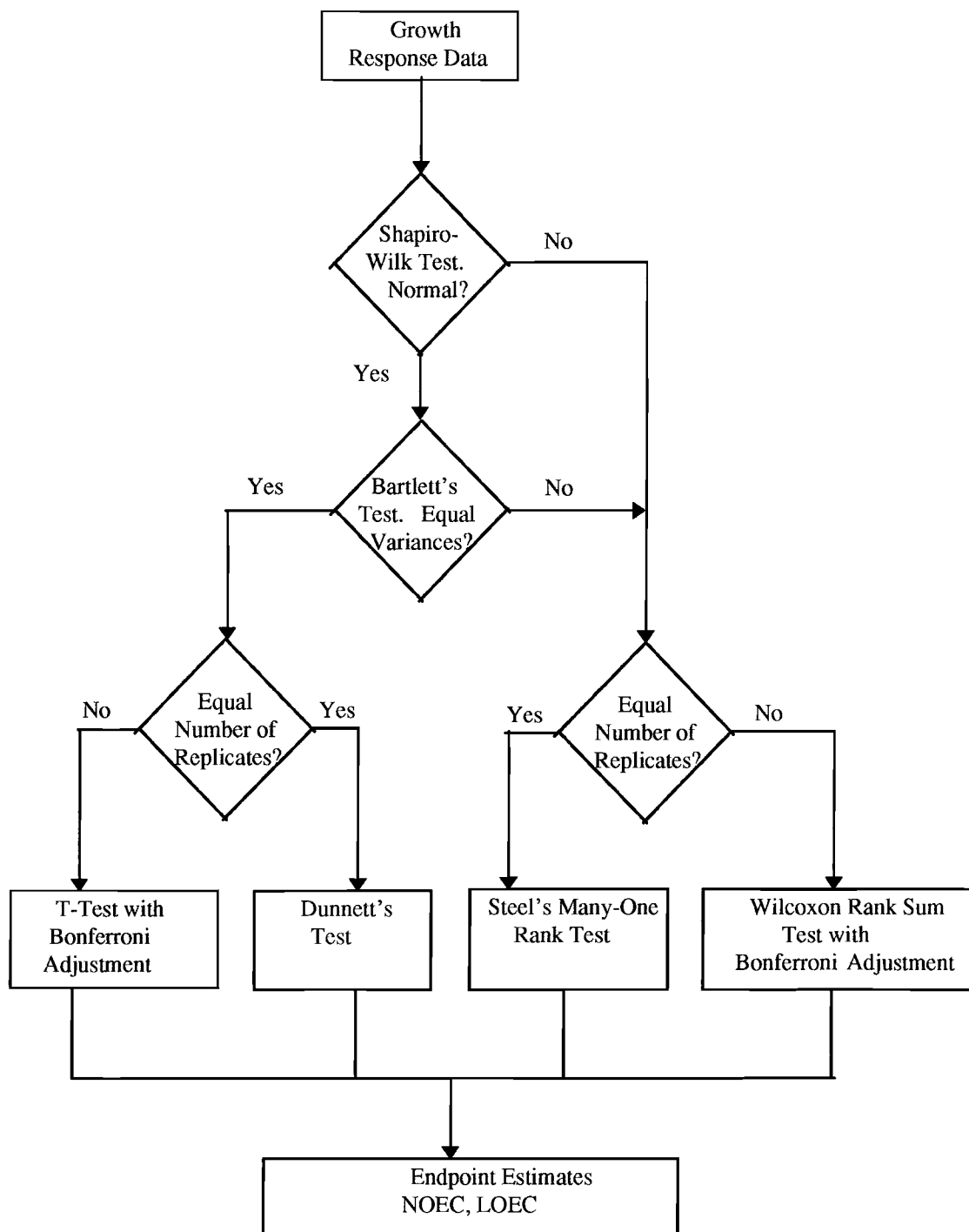


Figure 6. Flow Chart for Statistical Analysis of Algal Growth Response Data

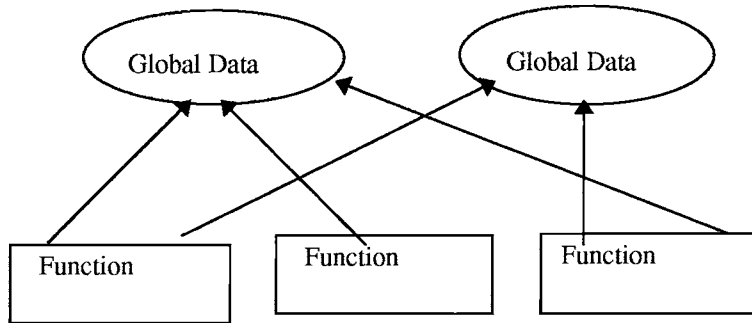


Figure 7. The Procedural Paradigm.

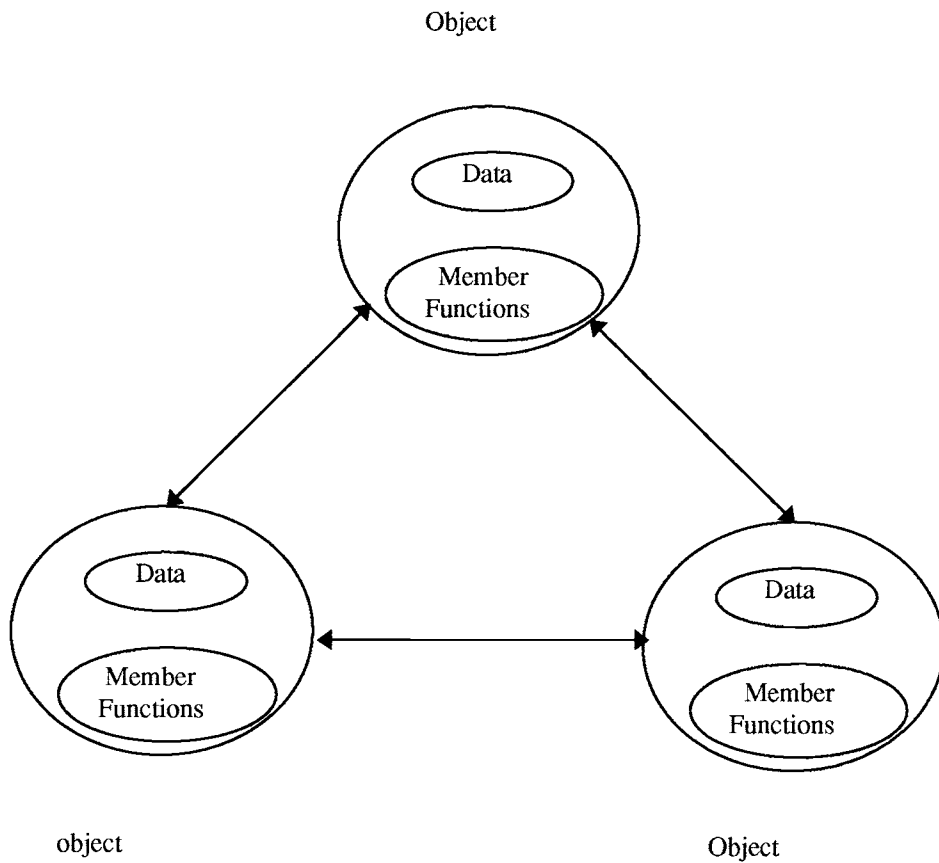


Figure 8. The Object-oriented Paradigm.

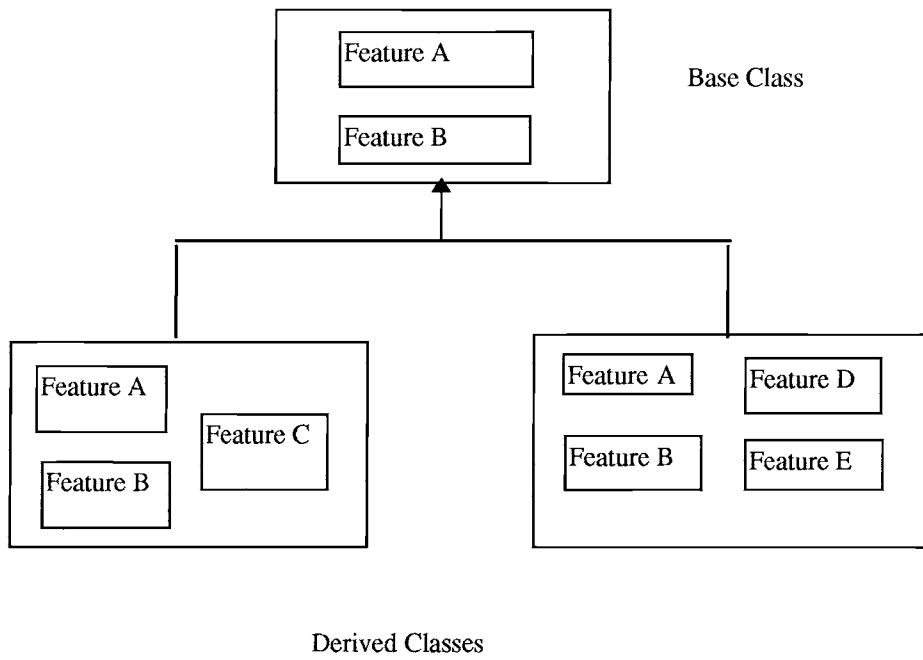


Figure 9. Inheritance.

CHAPTER III

DESIGN OF THE STATWIN PROGRAM

The Internal Design Overview

The author has designed and implemented the program in Microsoft Visual C++ 1.0 [Microsoft 1993]. From the flow charts in Figure 1 to Figure 6, it is clear that each method consists of several statistical tests, statistically each of which shares something in common. For example, testing data, the average of the data, or the variance of the data are common values needed for a statistical test. From the object-oriented program perspective, these common variables can be a base class of other derived classes.

The Graphical User Interface (GUI) is implemented as a Multiple Document Interface (MDI) window which includes a tool bar, a scroll bar which separates the data entry and graphical plot of data, dialog boxes, and control buttons etc.; it provides a convenient way for the users to select a desired dialog box for data entry or to execute the desired command.

Internal Structure for Statistical Tests

STATWIN is a program with statistical methods and graphical user interface. As mentioned in the previous paragraph, the statistical methods for measuring the toxicity have a common flow chart except for the Ceriodaphnia survival test. The manner of inputting data which is necessary for all the statistical tests and displaying the result of various tests is a matter of choice. Therefore, the author discovers the idea that breaks the

whole STATWIN program into two pieces, statistical tests and graphical user interface as shown in Figure 10. Dividing the whole STATWIN project into two parts makes it easier to focus on either statistical tests (ST) or the graphical user interface (GUI) one at a time. The communication between ST and GUI is by passing information necessary for ST from GUI and vice versa. If there are some changes needed to be made in ST, only local modification to ST is necessary because you know exactly what functions interact with it and leaves the GUI unchanged. Or if you want to revise the user interface, ST doesn't need to know the change of GUI as long as ST receives all information for the statistical calculation. By doing it in this way, it is easy to organize the structure and it helps maintain the integrity of STATWIN.

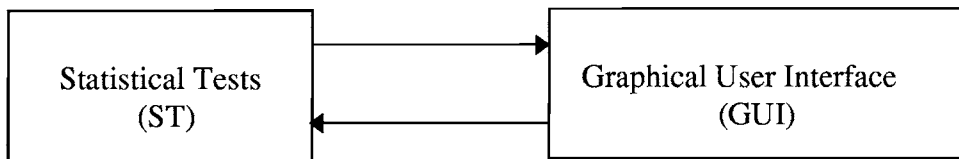


Figure 10. Internal Structure of STATWIN Program.

Statistical Tables Used in STATWIN

Seven statistical tables used in STATWIN have been implemented in the header files. The reason why tables aren't stored in binary files is that the author wants fast access to the tables. Binary files involve file I/O (input, output) which is much slower than the header files especially when the project gets large. Table III shows the relationship of statistical tests and their associated tables.

TABLE III
STATISTICAL TABLES AND TESTS

Name Of Statistical Test	Name of statistical table
Shapiro-Wilk Test	shwitbl.h
Bartlett's Test	barttbl.h
Dunnett's Test	dunntbl.h
T-Test with Bonferroni Adjustment	bonftbl.h
Steel's Many-One Rank Test	steeltbl.h
Wilcoxon Rank Sum Test With Bonferroni Adjustment	wilctbl.h
Fisher's Test	fishtbl.h

Transformation of the Data

According to the EPA document, data for some tests must first be transformed before being tested. Arc sine square root and \log_{10} are the two transformations actually being used in the EPA manual. As a result of the transformation of the raw data, only those data that are defined within certain intervals are eligible for a statistical test. Table IV shows any transformations used in the statistical test and the defined intervals for the raw data collected from a biological test.

TABLE IV
TRANSFORMATIONS AND STATISTICAL TESTS

Name Of Statistical Analysis	Transformation	Significant Interval For Raw Dada
Fathead Minnow Larval Survival	Arc Sine Square Root	[0, 1]
Fathead Minnow Larval Growth	N/A	N/A
Fathead Minnow Embryo-Larval	Arc Sine Square Root	[0, 1]
<u>Ceriodaphnia</u> Survival	N/A	N/A
<u>Ceriodaphnia</u> Reproduction	N/A	N/A
Algal Growth Response	\log_{10}	[10, $+\infty$]

The Statistical Tests

The C++ Classes Implemented

Nine ST related C++ classes have been implemented including BaseClass, ShapiroWilkClass, DunnettClass, BonferroniClass, BartlettClass, SteelClass, WilcoxonClass, InterpolationClass, and FisherClass. Figure 11 shows the relationships between these classes.

The BaseClass Class The BaseClass is the base class for all the classes of the ST portion of the program except for FisherClass which is exclusive for Ceriodaphnia survival test. BaseClass contains data members and member functions in which most of them would be reused or overwritten in derived classes. This is the philosophy of inheritance,

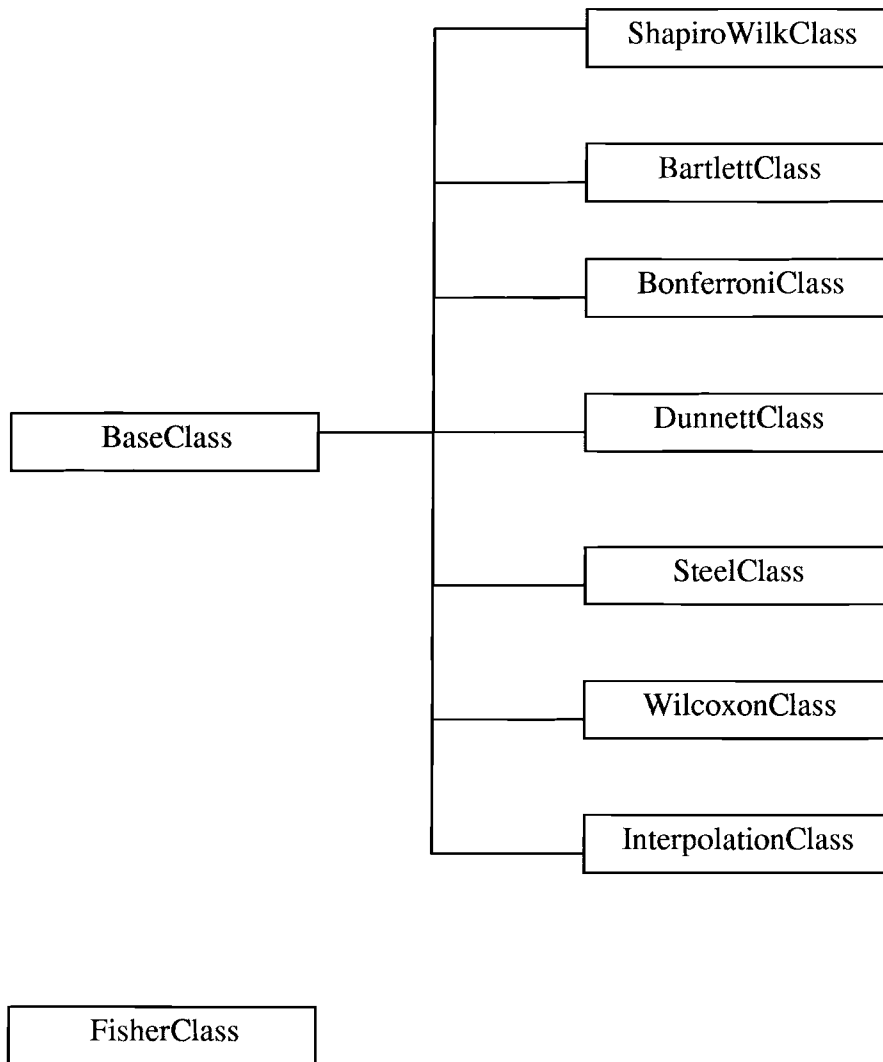


Figure 11. Class Hierarchy for Statistical Tests (ST).

one of the most important advantages of C++ language. Since inheritance permits code reusability which means after a base class is written and debugged, it need not be touched again, but can be adapted to work in different situations; it saves time and increases a program's reliability. In another words, if STATWIN needs to be extended later for some

reason, BaseClass is the first stop to visit for a programmer who wishes to add more statistical tests to the STATWIN.

For the data members of the BaseClass, such as two dimensional array of floating point variables for input data, p value of floating point variable for a statistical test, actual row size and column size used in the input data array, current position of the data, pointers to concentration levels, as well as pointers to mean and variance which are for derived classes of statistical tests, are the protect members of BaseClass. For the member functions, such as getInputData(), useTransformation(), getMean(), getVariance(), getGrandTotalObs(), etc. are the public member functions for derived classes.

The ShapiroWilkClass Class The ShapiroWilkClass class, derived from BaseClass, incorporates some new data members such as a pointer to float of ordered centered data, a pointer to float of coefficients for the test and members functions such as computeStatW(), SWgetCriticalValue() and Swtest() for Shapiro-Wilk test for normality. See Shapiro-Wilk test in Chapter II for a detailed explanation of the Shapiro-Wilk test.

The BartlettClass Class The BartlettClass class for equal variance is also derived from BaseClass. Besides all protected and public data members and member functions of BaseClass which can be adapted by the BartlettClass, the BartlettClass has contained data members, such as sum of degree of freedoms (df) of integer and pointer to integer of degree of freedoms for each concentration level and control, and member functions, such as computeStatB(), BgetCriticalValue() and Btest(). See Bartlett's test in Chapter II for a detailed explanation of Bartlett's test.

The DunnettClass Class The DunnettClass class, derived from BaseClass, has new data members and member functions for getting the interval of NOEC and LOEC. The examples of data members are pointer to integer for number of observations per concentration, total of sample observations of float, control mean of float, etc. Some of the member functions are computeSSB(), computeSST(), computeSSW(), computeMSB(), computeMSW(), computeStatT(), DgetCriticalValue(), and Dtest(). See Dunnett's test in Chapter II for a detailed explanation of Dunnett's test.

The BonferroniClass Class The BonferroniClass class is derived from BaseClass for parametric hypothesis testing when the number of replicates is not equal. Not only does it utilize all the protected and public members of BaseClass, but also it has its own data members and member functions. For example, the total of sample observations of float and pointer to integer of number of observations per concentration level are data members and computeSSB(), computeSST(), computeSSW(), computeMSB(), computeMSW(), BFgetCriticalValue() and Bftest() are member functions of BonferroniClass. See Bonferroni's test in Chapter II for a detailed explanation of Bonferroni's test.

The SteelClass Class The SteelClass class is also derived from BaseClass for non-parametric hypothesis testing for NOEC and LOEC when the number of replicates are equal for input data. Data members of SteelClass are pointers to float of group and ranked array. Member functions include SgetCriticalValue(), SassignRank(), SrankSum() and Stest(), etc. See Steel's test in Chapter II for a detailed explanation of Steel's test.

The WilcoxonClass Class The WilcoxonClass class is also derived from BaseClass for non-parametric hypothesis testing for obtaining interval of NOEC and LOEC when the number of replicates are unequal. The data members of WilcoxonClass are pointers to float of group and ranked array and member functions consist of WgetCriticalValue(), WassignRank(), WranksSum() and Wtest(), etc. See Wilcoxon's test in Chapter II for a detailed explanation of Wilcoxon's test.

The InterpolationClass Class The InterpolationClass, which is for point estimate, is a derived class of BaseClass because it is necessary to calculate mean value for each concentration level including control as well as the concentration levels input by the user in order to compute smoothed means for linear interpolation method. The data member is pointer to float of smoothed mean while member functions include IcomputeSmoothedMean(), IcomputeICp() and Itest(). See Linear Interpolation in Chapter II for a detailed explanation of Linear Interpolation.

The FisherClass Class The FisherClass, which is exclusive for Ceriodaphnia survival test, is independent of all above eight classes explained because the nature of the test is totally different from the other tests. This class is not designed as a base class; therefore no virtual functions are implemented. The stand alone class consists of private data member and public member functions. Examples of data members are integer of number of concentration, float of significantLevel, pointers to integer of dead, alive and total animal data and pointer to float of concentration levels, etc. Some of the member

functions are `getFisherInput()`, `getContingencyTable()`, and `Ftest()`, etc. See Fisher's test in Chapter II for a detailed explanation of Fisher's test.

Communication between ST and GUI

Because Fisher's test for Ceriodaphnia survival data is totally different from the rest of the statistical tests in the ways of both data entry and calculation, the connection between ST and GUI consists of two functions -- one for Fisher's test, `FisherConnection()`, and the other is `Connection()` for the rest of all tests. `Connection()` function is a two-way communication, so that data entered by user in graphical user interface is passed to ST and ST also returns results of tests to the GUI for graphical display if they are applicable. `FisherConnection()` function plays the same role of Ceriodaphnia survival test as `Connection()` function. Hence, ST and GUI are two modules of the STATWIN program, communicating with each other by passing information to and from via functions `Connection()` and `FisherConnection()`.

Graphical User Interface (GUI)

The Graphical User Interface (GUI) has been developed by using the Microsoft Foundation Class Library (MFC) which is part of Visual C++. MFC provides users the "application framework" which defines an architecture for integrating the user interface of an application for Windows with the rest of the application. It also provides the App Studio for constructing the user interface and editing resources, the Class Wizard for connecting Windows messages and user-interface objects such as menus to code.

The Document-View Architecture

The two most important parts of a Windows application are its document and its view. At Run time, a Windows application developed by using MFC is a group of cooperating objects that communicate with each other by sending and receiving Windows messages, calling each other's member functions. Documents are created by document template objects and managed by an application object (only one application object exists for a Windows application). Users interact with a document through a view object, which is framed by a document frame window object. Figures 12 shows graphically the relationships between document and view [Microsoft Visual C++, 1994].

The C++ Classes Implemented

Seven GUI related C++ classes have been implemented, which are CStatWinApp, CStatWinDoc, CMainFrame, CStatWinFrame, CGridEntry, CChartView and CAdmDlg. Figure 13 shows the relationships between these classes.

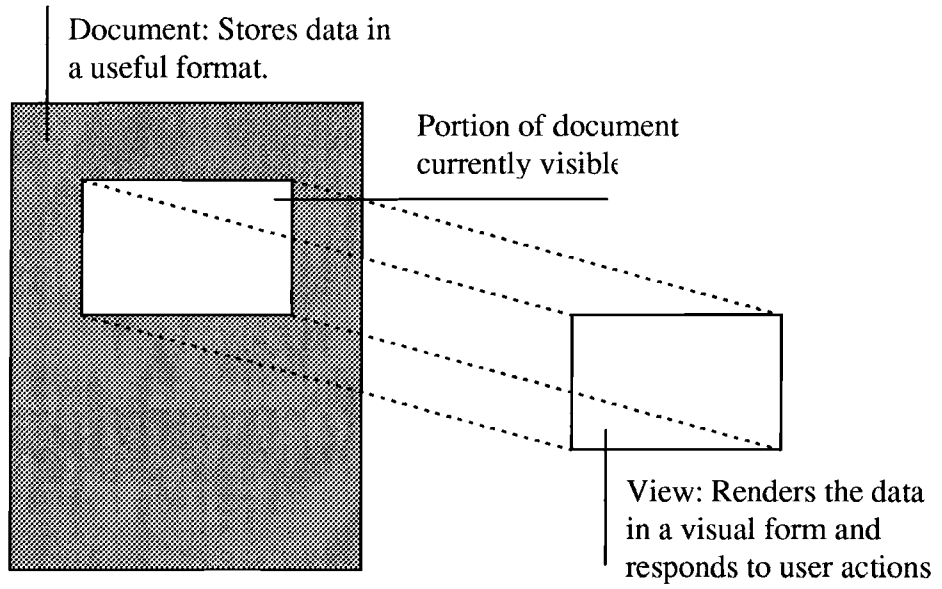


Figure 12. Document and View

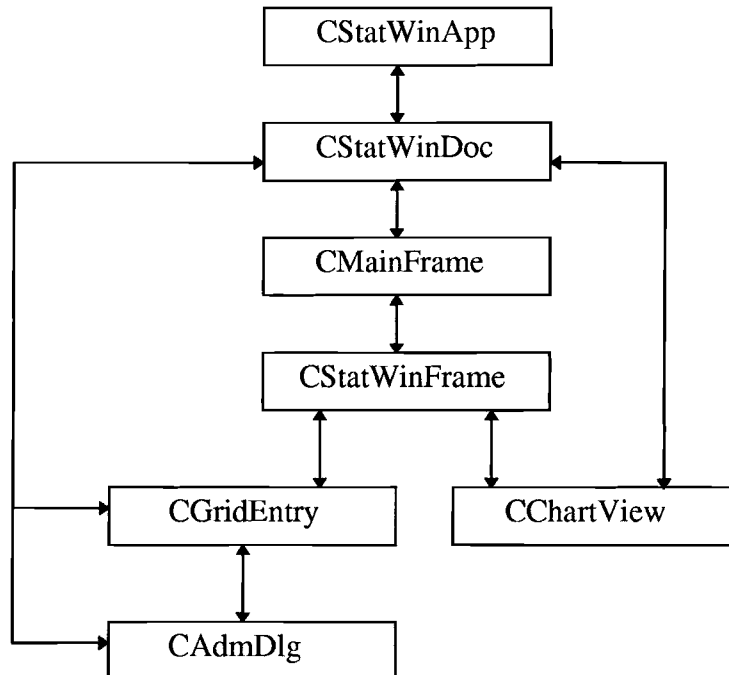


Figure 13. Internal structure of GUI for STATWIN.

The CStatWinApp Class The CStatWinApp class is derived from CWinApp which is one of the classes in MFC. CWinApp provides member functions for initializing and running Windows application. Each Windows application that uses MFC can only contain one object derived from CWinApp. The object is constructed when other C++ global objects are constructed and is available when Windows calls the WinMain (the counterpart of main in C program) function which is supplied by MFC. CStatWinApp overwrites the InitInstance member function to create the STATWIN application's main window object, and to register document templates which serve as the connection between documents (CStatWinDoc) and views (CGridEntry and CChartView).

The CStatWinDoc Class The CStatWinDoc Class is derived from CDocument which is one of the classes in MFC. CDocument provides the basic functionality for user-defined document class (a document represents the unit of data that the user typically opens with File Open command and saves with the File Save command). Users interact with the document through the CView object (CGridEntry) associated with it. CStatWinDoc defines the data member to store all user input data such as test type, analysis type, P value, test date, etc.; it also provides the functions for saving all user input data to a file and loading saved data from a file to STATWIN application.

The CMainFrame Class The CMainFrame Class is derived from CMDIFrameWnd which is one of the classes in MFC. CMDIFrameWnd provides the basic functionality of a multiple document interface (MDI) frame window and members for managing the window. The CMainFrame provides the member functions for executing the users

specified statistics test (the “EX” button) and for opening the report (the “R” button). It also provides the droplist for users to select the plot type (Line, 2D Bar or 3D Bar). See Figure 14.

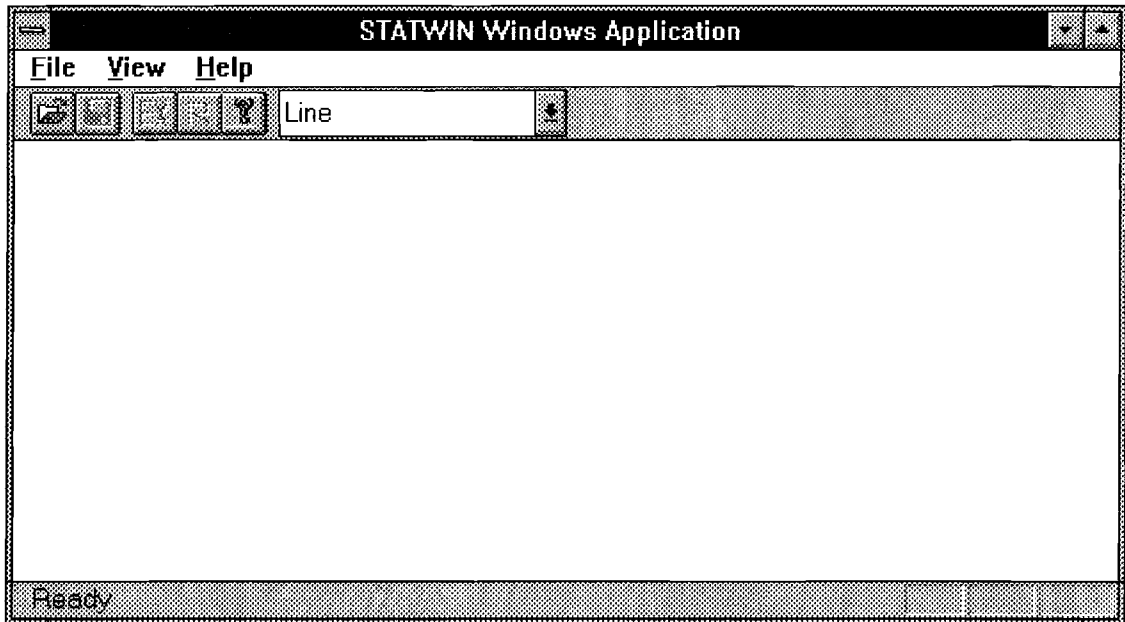


Figure 14. Main Frame Window of STATWIN

The CStatWinFrame Class The CStatWinFrame Class is derived from CMDIChildWnd which is one of the classes in MFC. CMDIChildWnd provides the basic functionality of a Multiple Document Interface (MDI) child window and members for managing the window. An MDI child window looks like a frame window, except that it appears inside an MDI frame window; it does not have a menu bar of its own, but instead shares the menu of the MDI frame window. CStatWinFrame divides the window into two views (splitter window); the left view (CGridEntry) is used for users to enter input data,

and the right view is used for plotting user input data and calculation results. See Figure 15.

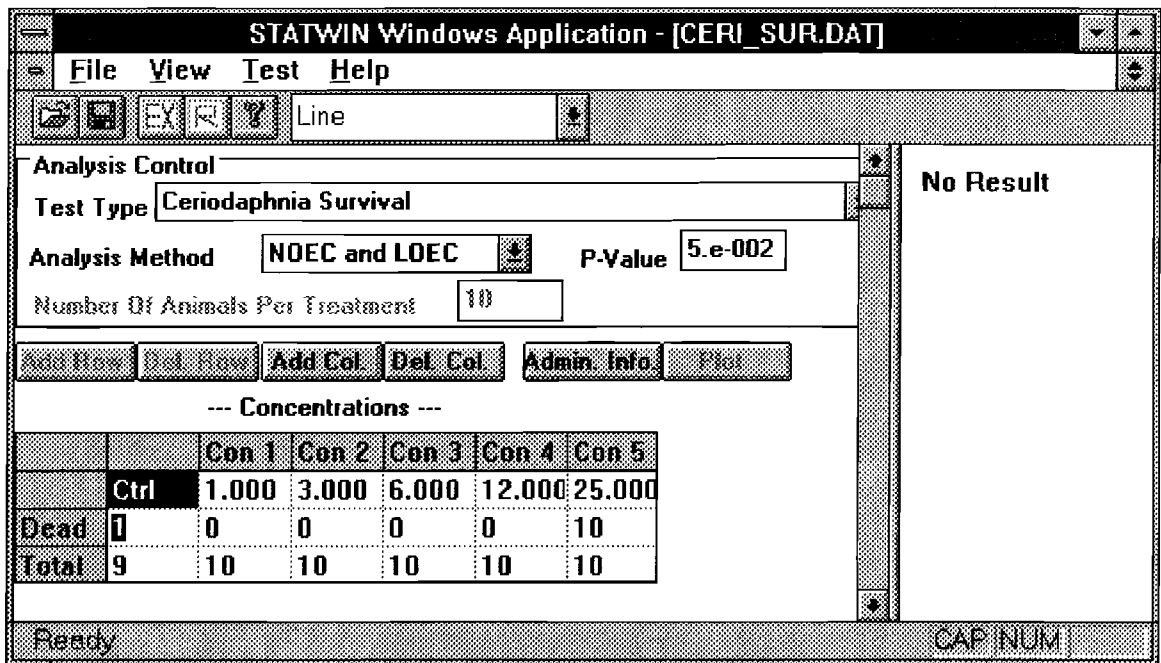


Figure 15. Splitter Window of STATWIN

The CGridEntry Class The CGridEntry Class is derived from CFormView which is one of the classes in MFC. CFormView is the base class for viewing controls (these controls are created by App Studio). CGridEntry contains the following controls:

- 1) Test Type: Droplist for user to select a specific test type.
- 2) Analysis Method: Droplist for user to select a specific analysis method.
- 3) P Value: Edit control for user to enter the specific P value.
- 4) Number Of Animals Per Treatment: Edit control for user to enter the specific value.
- 5) Grid: Visual Basic control for user to enter data in a table format.

- 6) Add Row: Button control. When users press the button, a new row is added to the grid and the maximum number of rows allowed is 15.
- 7) Del. Row: Button control. When users press the button, the last row is deleted from the grid and the minimum number of rows required is 3.
- 8) Add Col.: Button control. When users press the button, a new column is added to the grid and the maximum number of columns allowed is 10.
- 9) Del. Col.: Button control. When users press the button, the last column is deleted from the grid and the minimum number of columns required is 3.
- 10) Admin. Info.: Button control. When users press the button, a dialog window will show up.
- 11) Plot: Button control. When users press the button, the calculation results (if applicable) will be plotted in the right view. See Figure 16.

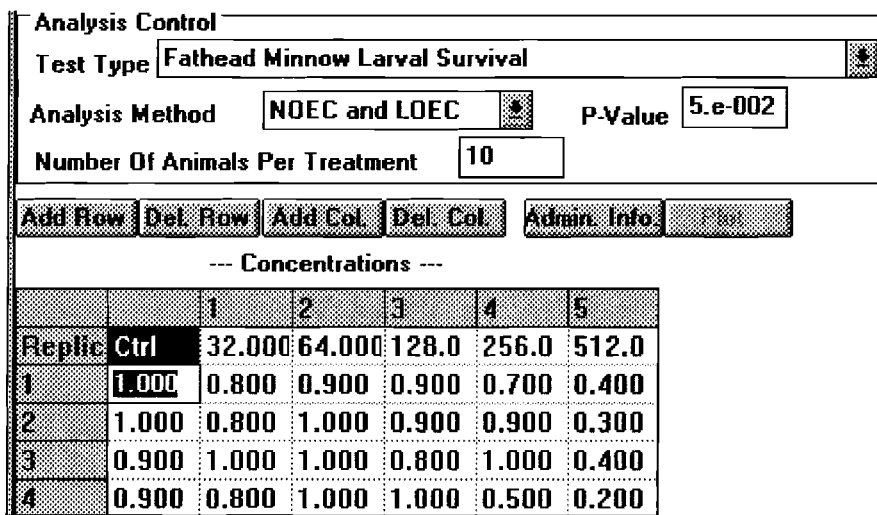


Figure 16. Data Entry of STATWIN.

The CChartView Class The CChartView Class is derived from CView which is one of the classes in MFC. CView provides the basic functionality for user-defined view classes. A view is attached to a document and acts as an intermediary between the user and the document. CChartView provides functions for creating line plot, 2D Bar plot and 3D Bar plot. Users can select a specific type of plot through the droplist on the toolbar. Figure 17 gives line display.

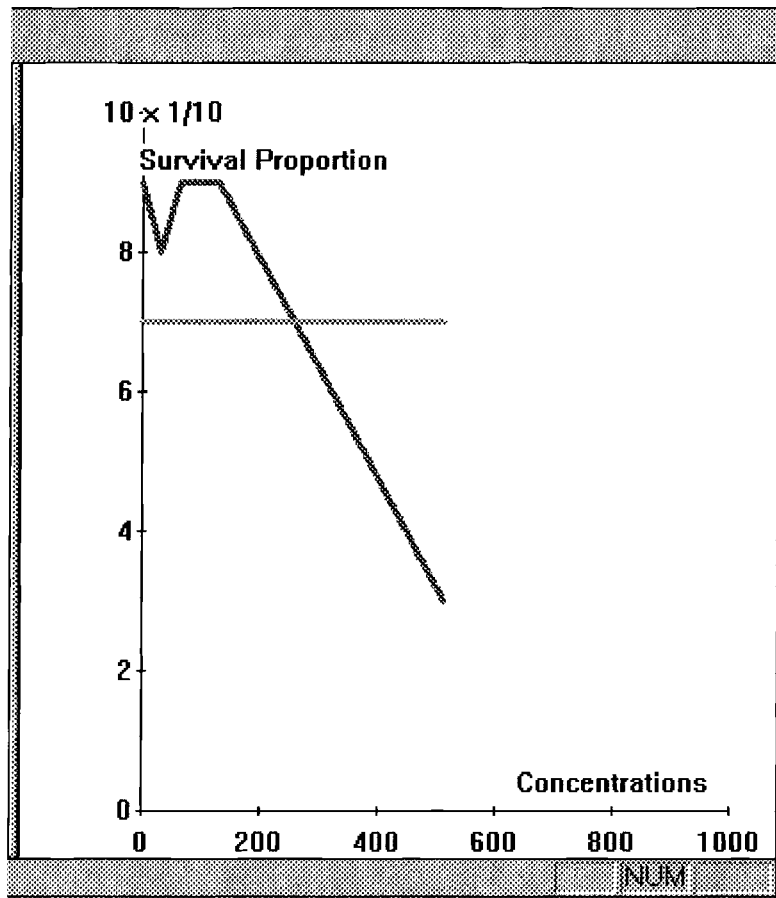


Figure 17. Plot of STATWIN.

The CAdmDlg Class The CAdmDlg Class is derived from CDialog which is one of the classes in MFC. CDialog is the base class for displaying dialog boxes on screen. The CAdmDlg Class is used for users to enter the administrative information of the statistics test. This information includes Discharger, Location, Test Date and Analyst. Figure 18 shows the dialog box.

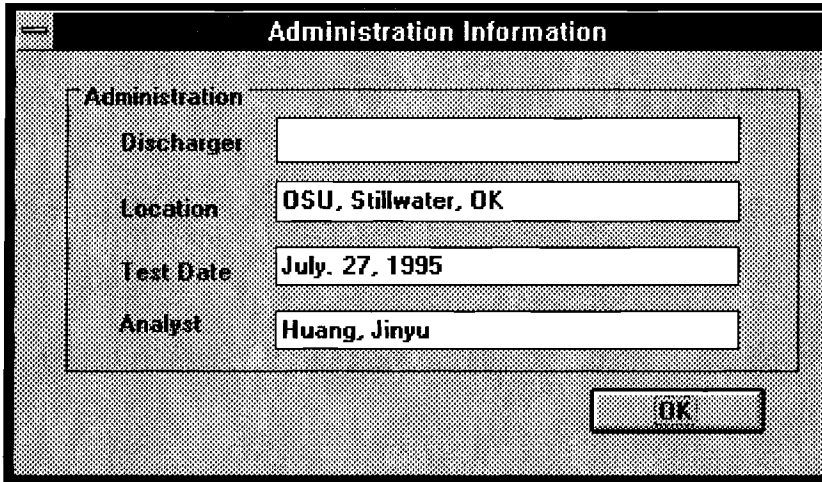


Figure 18. Administration Information Dialog Box of STATWIN.

CHAPTER IV

CONCLUSION

Summary of STATWIN and Major Improvements

STATWIN is a computer implementation of the short-term methods for measuring the chronic toxicity of effluents on freshwater organisms which was created by the United States Environmental Protection Agency (EPA) in 1989. Moreover, the Graphical User Interface (GUI) has been developed in STATWIN in terms of pull-down menus, toolbar, droplist, control buttons and message box windows, etc. by using Microsoft Foundation Class Library (MFC). The object-oriented paradigm has been employed for either the internal structure of statistical tests or the graphical user interface of the program. In addition, the Document - View architecture in Microsoft Visual C++ [Kruglinski 1993], in which data is stored in a document and displayed in a view and by sending and receiving windows messages, document and view communicates with each other, plays a key role in the STATWIN program.

There are several advantages of using STATWIN for short term methods to estimate toxicity on freshwater organisms.

First, users don't have to be statistics experts in order to analyze data collected from biological tests when following the instructions of EPA manual. Instead, STATWIN automates the series of statistical tests to help users to conduct statistical analysis on raw data. Thus, much of the time and attention can be paid to biological tests rather than statistical tests without human errors while calculating the statistical tests.

Secondly, STATWIN is user friendly with interactive communication with users. For example, users are able to select six Test Types and two Analysis Methods which are provided by the STATWIN. Besides, users not only decide how accurate they want their analysis, but also they can customize their need of the size of conducting their biological tests. By continuing to click Add Row, Del. Row, Add Col., and Del. Col. buttons, users are able to get an exact size of the table for all the input data.

Thirdly, after a statistical test has been done, users can read a report of the test currently being analyzed by simply clicking the R button in toolbar. The report has full features of a file in Windows because it is a Notepad file in Microsoft Windows 3.1. Therefore, users can open, save and print a file, copy, paste and search a text in the file etc. Report also lists the results of each statistical test performed on data.

Finally, graphical plots are produced in this version of STATWIN in three ways, which are line, 2D bar and 3D bar. Using an example of a fathead minnow larval survival test, the curve in red in the color monitor connects the mean value for each concentration including control mean while the curve in green represents the critical value for Dunnett's test. From the graphical plots, it's easy to see that any proportion below the green curve would be significantly different from the control as shown in Figure 17. Thus, graphical plots are a good aid in interpretation of data and results.

Possible Extension of STATWIN

Currently, STATWIN allows users to enter up to 15 rows and 10 columns of data because the maximum numbers of row size and column size are hard coded as 15, and 10

respectively in the program. In order to increase the row size and column size, users can change the values in constant.h file and rebuild the affected files in the Microsoft Visual C++ workbench. Although it's easy to increment row size and column size of STATWIN, one should know that when the sample size (row size in STATWIN) exceeds 50, the Shapiro-Wilk test for normality is not appropriate. The Kolomogorov "D" statistic is recommended by the EPA.

The objective of chronic aquatic toxicity tests with effluents is to estimate the highest "safe" level of these substances, so two statistical methods which are hypothesis testing and point estimate are used in the EPA manual. No-Observed-Effect-Concentration (NOEC) and Lowest-Observed-Effect-Concentration (LOEC) are determined by hypothesis testing, whereas Effective Concentration (EC), Lethal Concentration (LC) and Inhibition Concentration (IC) are determined by point estimates. Of the two point estimations Linear Interpolation and Probit Analysis, only Linear Interpolation for IC is available in STATWIN.

With the object-oriented paradigm used in STATWIN, it should be not difficult to add more classes for possible tests as mentioned in the previous paragraph. One of the greatest advantages of the object-oriented approach to the problem is local modification to the program which allows the author to make changes to the local area without touching another part of the program, so it's easy to organize and maintain the integrity of the program. Besides the local modification, the author also makes use of one of the best features of an object-oriented language like C++ -- inheritance. Common operations for a statistical test such as mean and variances are already implemented in the base class.

Therefore, new classes for new statistical tests can be derived from the base class which reduces the code redundancy and workload of future work.

BIBLIOGRAPHY

- Bain, L. J. and Engelhardt, M. (1987). Introduction to Probability and Mathematical Statistics. Duxbury Press, Boston.
- Introducing Visual C++. (1994). Microsoft Corporation.
- Kruglinski, D. J. (1993). Inside Visual C++. Microsoft Press.
- Lafore, R. (1991). Object-oriented Programming in Turbo C++. Waite Group Press.
- Microsoft Visual C++ User's Guides (1993). Microsoft Corporation.
- Ott, L. (1988). An Introduction to Statistical Methods and Data Analysis. PWS-KENT Publishing Company, Boston.
- Smith, K. L. (1991). TOXY: A Statistical Program to Analyze the Toxicity of Water Samples on Freshwater Organisms. M. S. Thesis, Oklahoma State University, Stillwater, Oklahoma.
- Weber C. I., et al. (1987). "Short-term Methods for Estimating the Chronic Toxicity of Effluents and Receiving Waters to Marine and Estuarine Organisms". U. S. Environmental Protection Agency, Environmental Monitoring Systems Laboratory, Cincinnati, OH, EPA/600/4-87/028.
- Weber C. I., et al. (1989). "Short-term Methods for Estimating the Chronic Toxicity of Effluents and Receiving Waters to Freshwater Organisms". U. S. Environmental Protection Agency, Environmental Monitoring Systems Laboratory, Cincinnati, OH, EPA/600/4-89/001.

APPENDIX

APPENDIX

STATWIN USER'S MANUAL

INTRODUCTION

STATWIN is an object-oriented approach to estimate the toxicity of effluents on freshwater organisms for the U. S. Environmental Protection Agency document "Short-term Methods for Estimating the Chronic Toxicity of Effluents and Receiving Waters to Freshwater Organisms".

STATWIN provides end users a graphical user interface (GUI) environment in terms of pull-down menus, dialog boxes, and drop lists etc. Users, even with limited knowledge of statistics, are able to select a specific statistical analysis for data collected from a biological test suggested in the EPA document without worrying about the detail sequence of statistical tests performed.

Users are also given the right to determine how accurate they want the tests being done by entering the p-value, which gives the confidence interval of the test. Additionally, users can also obtain graphical displays of the data after the statistical test is done.

The requirement to run the STATWIN program is Windows 3.1 and a color monitor.

GENERAL DESCRIPTION OF THE PROGRAM

STATWIN.EXE is a STATWIN executable file which runs in the Microsoft Windows 3.1. If users run the executable file in DOS accidentally, DOS gives the error message and asks users to run it in the Windows environment.

Testing data files have been provided in the same directory as STATWIN.EXE. The table V shows the content of the testing data files.

TABLE V
A SUMMARY OF TESTING DATA IN THE STATWIN PROGRAM

Data File	Description of the testing data file
larv_sur.dat	Fathead minnow larval survival data.
larv_gth.dat	Fathead minnow larval growth data.
larv_emb.dat	Fathead minnow embryo-larval data.
ceri_sur.dat	<u>Ceriodaphnia</u> survival data.
ceri_rep.dat	<u>Ceriodaphnia</u> reproduction data.
algl_gth.dat	Algal growth response data.
larv_gpt.dat	Fathead minnow larval growth data for point estimate.
ceri_rpt.dat	<u>Ceriodaphnia</u> reproduction data for point estimate.

STARTING AND QUITTING STATWIN

There are three ways to run STATWIN in Windows 3.1.

To Start STATWIN

You can start STATWIN by doing one of the three ways.

1. In the File Manager, find STATWIN.EXE and double click it. STATWIN starts and displays windows of data entry, which is STATWIN Windows Application.
2. In the Program Manager, from the File menu, choose Run. Type the correct path followed by STATWIN.EXE to the command line and click OK button.
3. Double click the Microsoft Visual C++. From the Project menu, choose Open and type the correct path and project name. Build and execute the project.

The first two ways require no source files of STATWIN, nor Microsoft Visual C++ tool, only the executable file STATWIN.EXE. The third way of starting STATWIN needs all the source files of STATWIN and Microsoft Visual C++. No matter which way you select to use, the STATWIN Windows Application - [StatWin1] displays on the screen as Figure 19 shown.

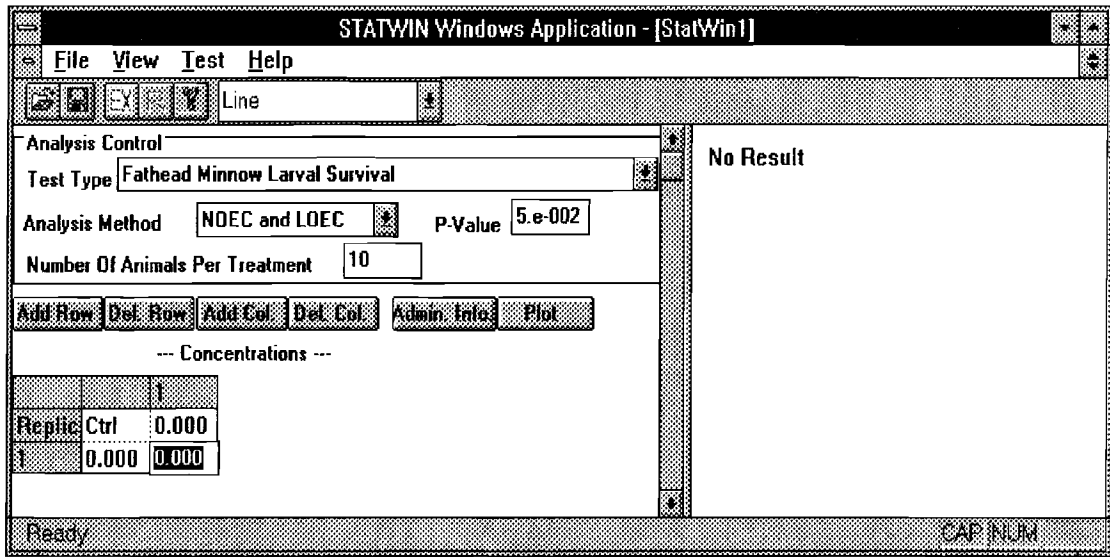


Figure 19. Start Up Window of STATWIN

To Quit STATWIN

From the File menu, choose Exit.

DATA ENTRY AND EXECUTION OF THE PROGRAM

After the STATWIN.EXE runs, the Figure 19 shows up.

The STATWIN Windows Application - [StatWin1] is a splitter window, which consists of two parts. On the left side of the window is the data entry area. On the right side is the graphical display of the test results.

Raw data is inputted in the grid of the left view where the first row is fixed for sequence number of concentration levels and the first column is fixed for sequence number of replicate for different concentrations. Starting from the second row, third column, one should enter concentration levels. The values in the second column are from control treatment which is used to compare to each concentration level.

Before you are about to experience STATWIN, you also need to know the graphical plot of the statistical analysis for data collected from a biological test. The graphical plot won't display on the screen until after a statistical analysis is done and Plot button is clicked.

To Input Test Data other than Ceriodaphnia Survival Test

1. Click Add Row button and/or Add Col. button to input test data and concentration levels in the grid area which is a Visual Basic control for user to enter data in a table.
2. Select one of the tests in the droplist Test Type. There are 6 analysis tests in the droplist as shown in Figure 20.

3. Select one of the analysis methods in the droplist Analysis Method which contains two analyses as in Figure 21.
4. Type the p-value necessary for a test in P-Value.
5. If your Test Type is other than Ceriodaphnia Survival and Analysis Method is NOT NOEC and LOEC, you need to give a value to Number Of Animals Per Treatment.
6. If your test is Ceriodaphnia Survival and Analysis Method is NOEC and LOEC, you follow the instruction of how to enter data for Ceriodaphnia Survival test.

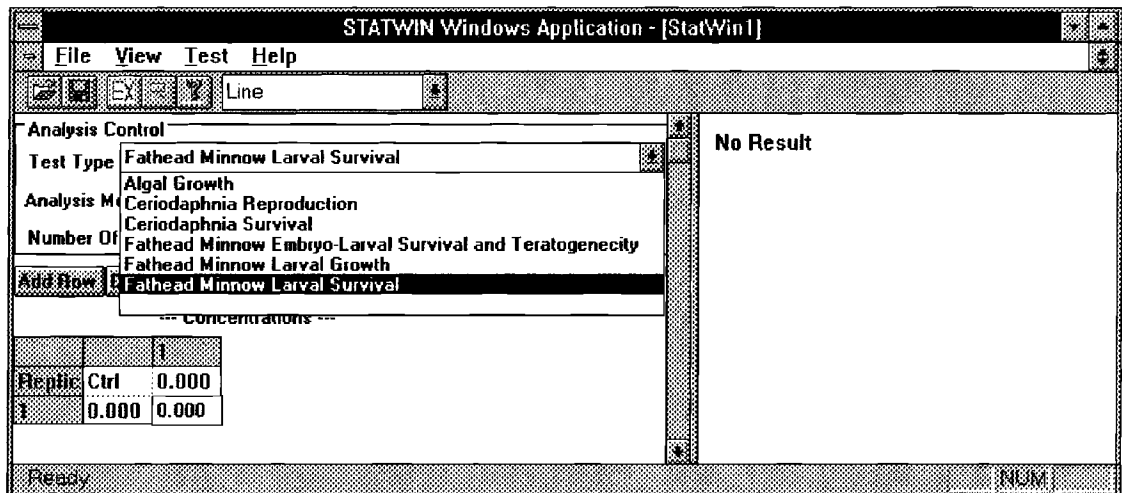


Figure 20. Available Selections of Test Type

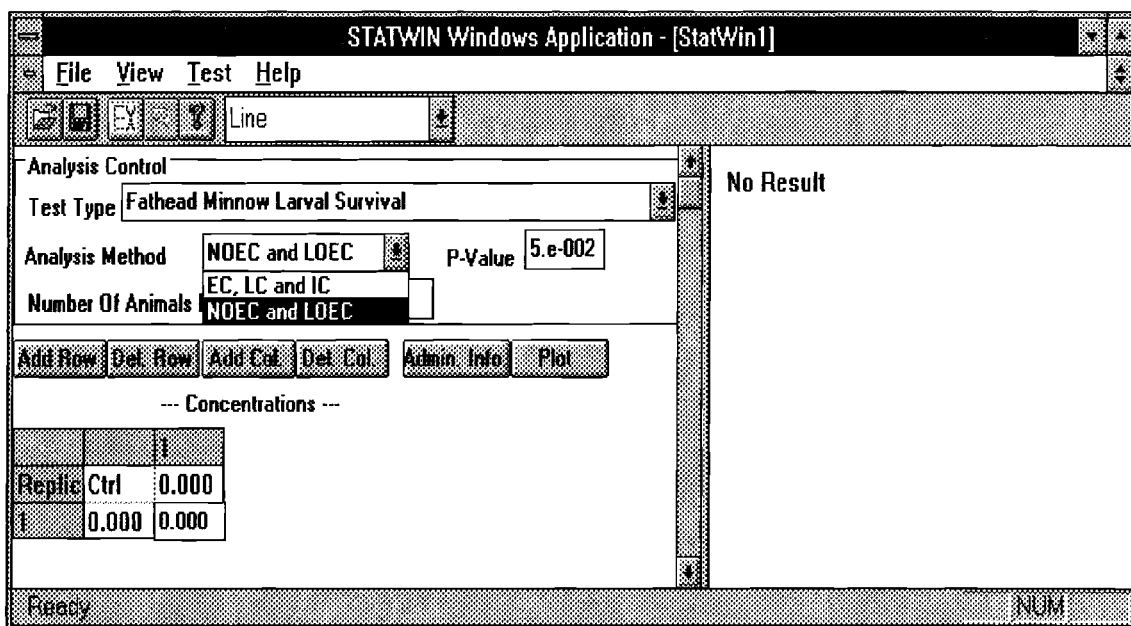


Figure 21. Available Selections of Analysis Method

To Input Test Data for Ceriodaphnia Survival Test

1. Select Test Type Ceriodaphnia Survival.
2. Select Analysis Method NOEC and LOEC.
3. Type a value for P-Value.
4. Notice Number Of Animals Per Treatment is grayed out which means this option is disabled and you don't need to type anything.
5. Notice Add Row and Del. Row are disabled because this test requires only two rows and as many columns as necessary based on the biological tests.

See Figure 22.

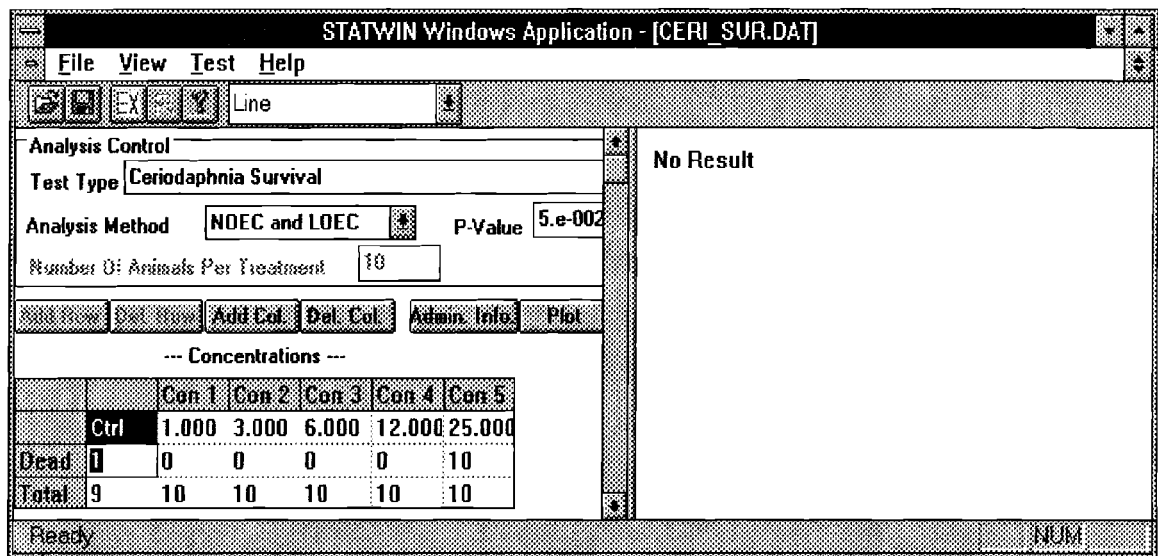


Figure 22. Example Input for Ceriodaphnia Survival Test

To Enter Administration Information

1. Click the Admin. Info. button, the administration information dialog box will show up as in Figure 23, and type any administration information. Click OK button to get back to STATWIN Windows Application - [StatWin1].

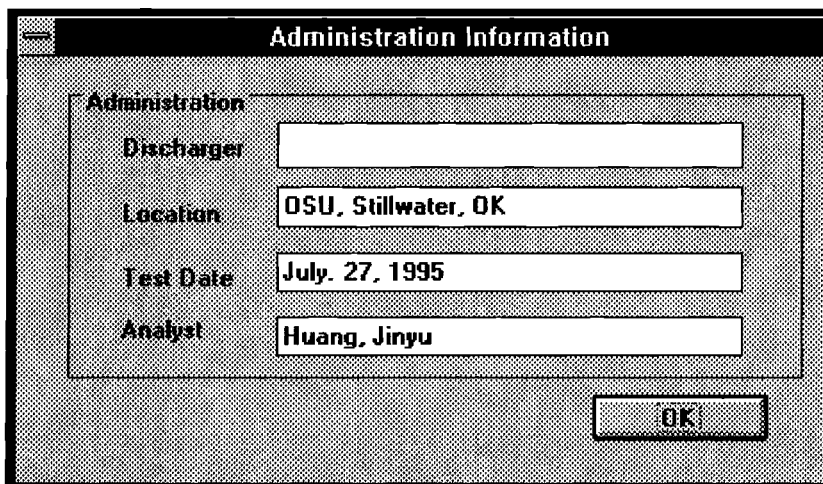


Figure 23. Administration Information Dialog Box

To Save All Data in the STATWIN Windows Application - [StatWin1]

You can either

1. Click Save button in tool bar as shown is Figure 24. It takes you to Figure 25.

Give the valid path in Directories and name in File Name. Click OK button and data will be saved in a file.

or

2. From the File menu, choose Save As or Save menu, it also takes you to the Figure 25.

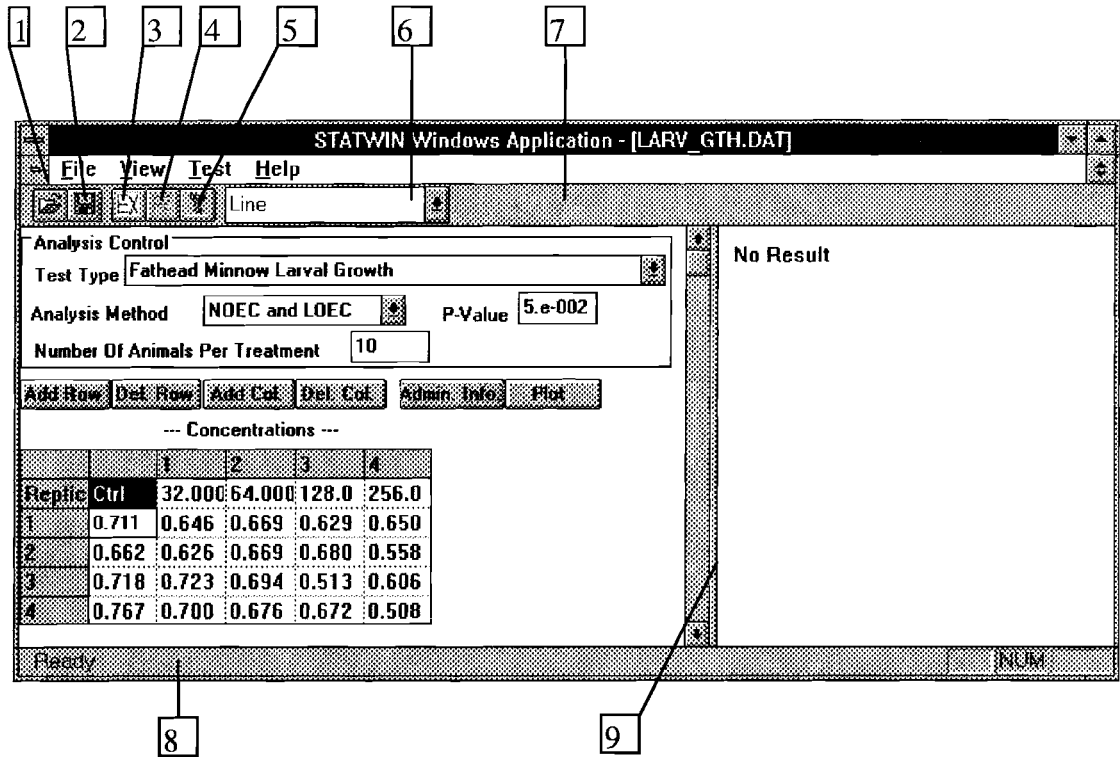


Figure 24. Tool Bar and Status Bar

1. Open file button.
2. Save file button.
3. Execution button.
4. Report button.
5. Help button.
6. Graphical plot selection button.
7. Tool bar.
8. Status bar.
9. Scrolled bar.

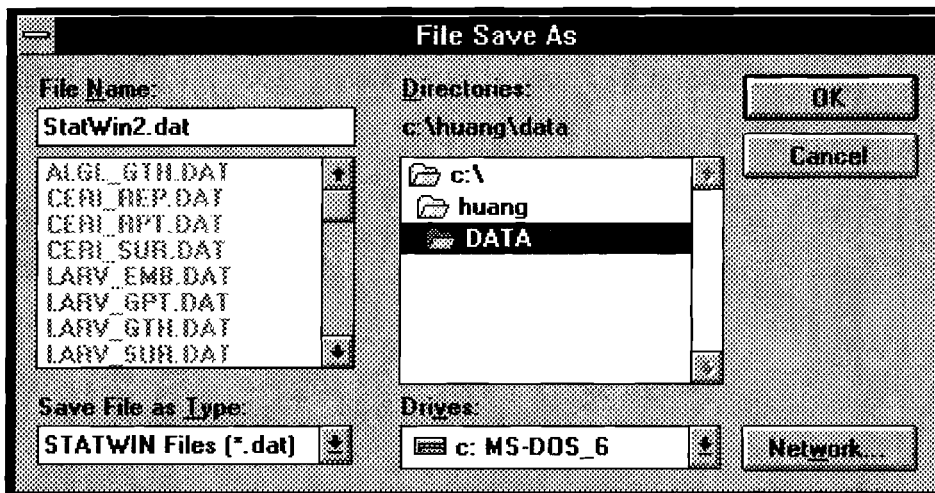


Figure 25. File Save As Dialog Box

To Execute the Program

You can either

1. Click the EX button.

or

2. From the Test menu, choose Execute menu.

then,

3. the program executes. After the program is executed, there is a Message Box popping up as shown in Figure 26. Click OK button.



Figure 26. Execution Message Box

To See the Report of the Current Test

You can either

1. click R button which stands for report in the tool bar

or

2. From the Test menu, choose Report.

3. Figure 27 displays on the screen. It's the standard Notepad in the Microsoft Windows 3.1.

To See the Graphical Plot for the Test Results

1. Click Plot button.
2. The graphical plot displays on the right view separated by a scrolled bar from the data entry, left view as Figure 28 shows.
3. You have three ways of graphical display to choose from. The default is line.
4. To choose graphical display other than line representation, click the last button in tool bar as Figure 28 shows. You can select 2D Bar or 3D Bar.
5. If click 2D Bar, then Figure 29 shows.
6. If click 3D Bar, then Figure 30 shows.

More Information

The following is a brief explanation of the graphical plot.

After the execution of the program, if you click the Plot button, then the graphical plot shows up on the right view. As an example, see Figure 28. In a colored monitor, you see one red curve and one green curve.

Notepad - fishout.out

File Edit Search Help

-----Fisher's test for Ceriodaphnia survival -----

	Number of		Number of Observations
	Alive	Dead	
1% Effluent	10	0	10
Control	8	1	9
Total	18	1	19

	Number of		Number of Observations
	Alive	Dead	
3% Effluent	10	0	10
Control	8	1	9
Total	18	1	19

	Number of		Number of Observations
	Alive	Dead	
6% Effluent	10	0	10
Control	8	1	9
Total	18	1	19

	Number of		Number of Observations
	Alive	Dead	
12% Effluent	10	0	10
Control	8	1	9
Total	18	1	19

	Number of		Number of Observations
	Dead	Alive	
25% Effluent	10	0	10
Control	1	8	9
Total	11	8	19

FINAL RESULT FROM FISHER TEST FOR SIGNIFICANT LEVEL 0.05
NOEC = 12%
LOEC = 25%

Figure 27. Example of STATWIN Report

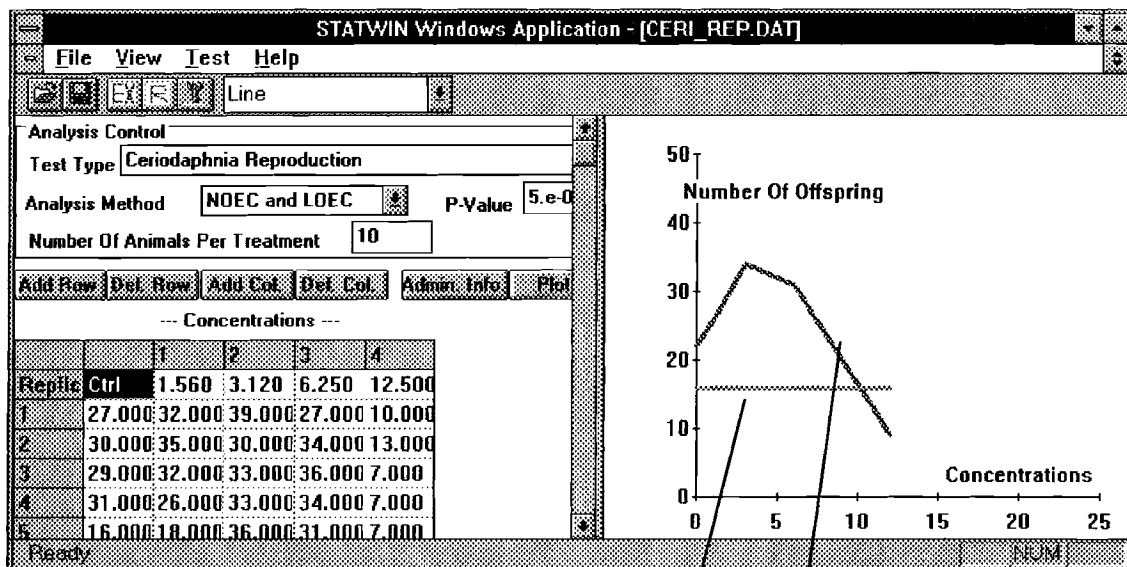


Figure 28. Example of Line Plot

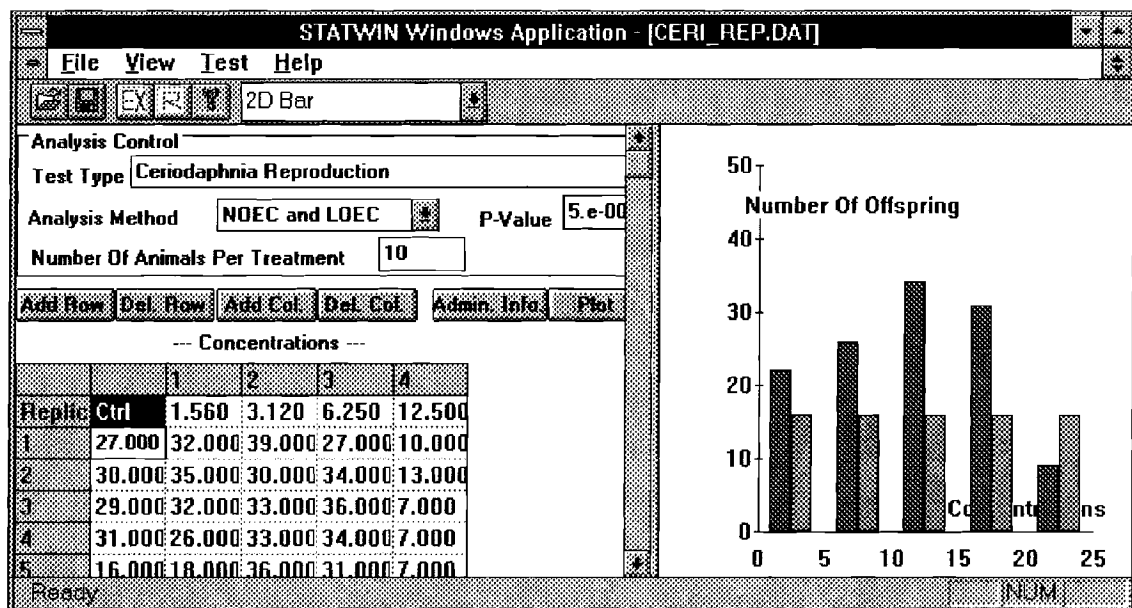


Figure 29. Example of 2D Bar Plot

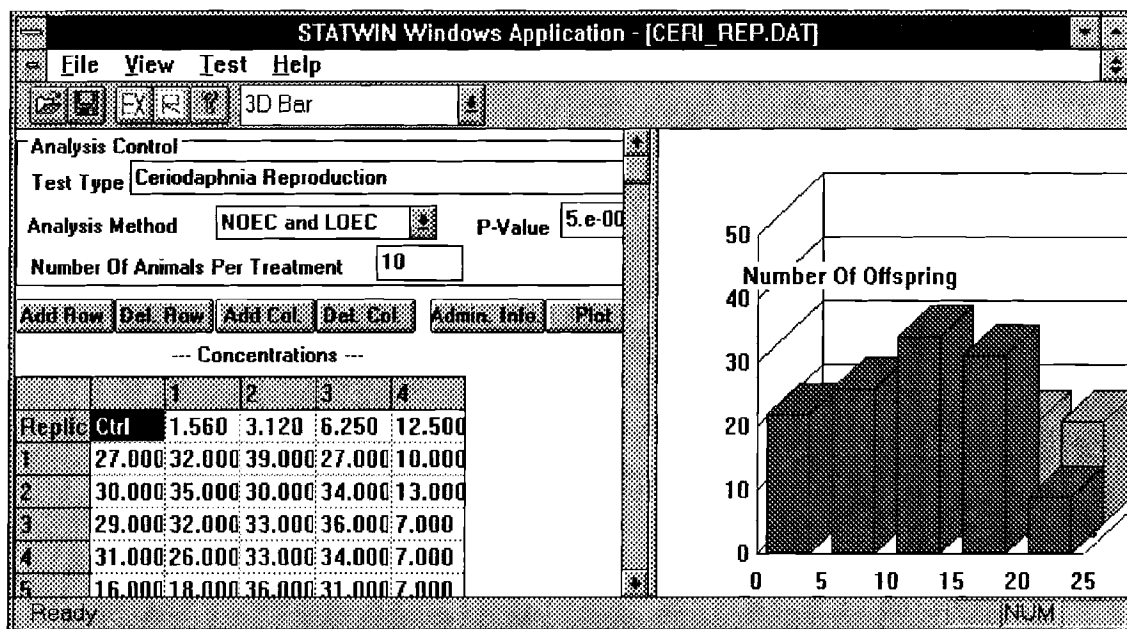


Figure 30. Example of 3D Bar Plot

The red curve connects the mean value for each concentration including control while the green curve represents the critical value for Dunnett's test which means any proportion below this value would be significantly different from the control in Figure 28. (This test is *Ceriodaphnia* Reproduction, NOEC and LOEC.)

To See Information Concerning STATWIN

You can either

1. Click ? button in tool bar

or

2. From Help menu, choose About STATWIN,

3. Then Figure 31 displays.

4. Click OK button.

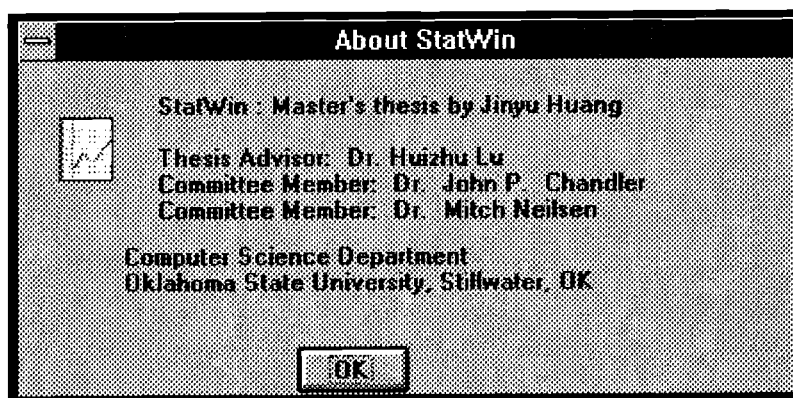


Figure 31. About STATWIN Dialog Box

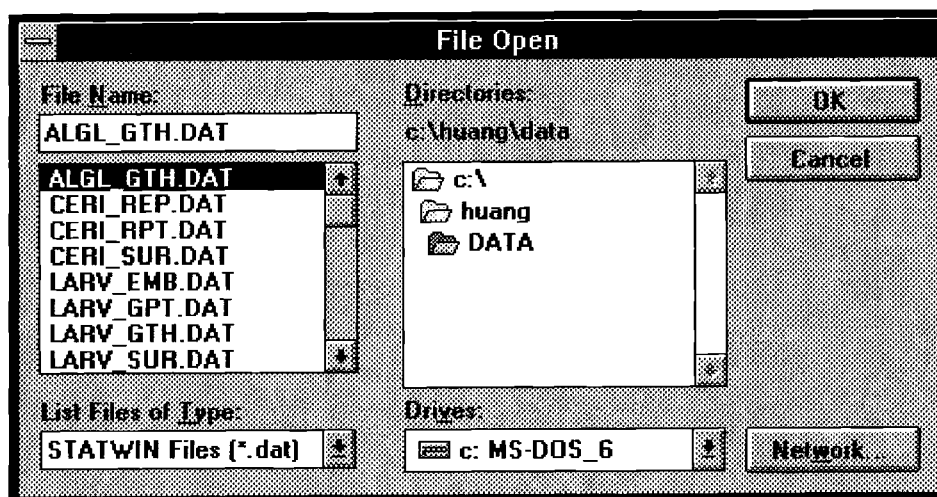


Figure 32. File Open Dialog Box

OPENING AN EXISTING TESTING DATA FILE AND EXECUTING THE FILE

To Open a File

You can either

1. click the first button in tool bar

or

2. From the File menu, choose Open.

3. The Figure 32 displays o the screen; select a proper directory in Directories and type an existing file name in File Name.

4. Click OK button to get a testing data file.

5. Follow the instruction of the section of To Execute the Program.

TOOLBAR AND STATUSBAR

ToolBar and statusBar displaying on the screen is the default option of the program.

To Hide or Display ToolBar

1. From the View menu, uncheck Toolbar, then toolbar disappears as shown in Figure 33.

2. From the View menu, check Toolbar, then toolbar appears on the screen.

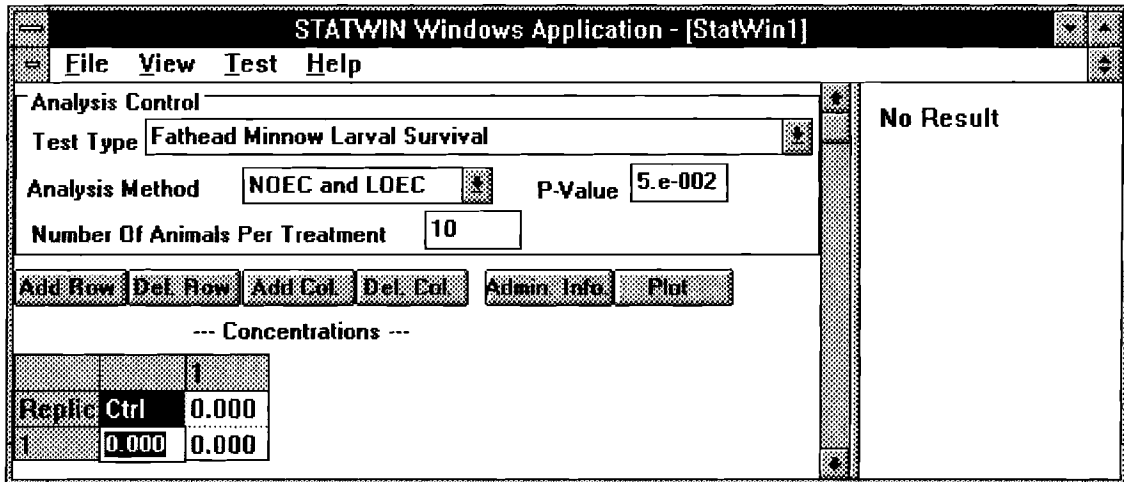


Figure 33. Frame Window without Toolbar and StatusBar

To Hide or Display StatusBar

1. From the View menu, uncheck StatusBar, then statusBar disappears as shown in Figure 33.

2. From the View menu, check StatusBar, then statusBar appears on the screen.

VITA

Jinyu Huang

Candidate for the Degree of

Master of Science

Thesis: STATWIN: AN OBJECT-ORIENTED APPROACH TO ESTIMATE THE TOXICITY OF EFFLUENTS ON FRESHWATER ORGANISMS

Major Field: Computer Science

Biographical:

Personal Data: Born in Wuhan, Hubei Province, P.R.China, on December 25, 1967, the daughter of Tianxi Huang and Huiying Liu.

Education: Graduated from Oklahoma State University, Stillwater, Oklahoma in July, 1992; received Bachelor of Science degree in Statistics. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in December 1995.

Experience: Graduate teaching assistant, Oklahoma State University, Department of Computer Science, August 1994 to December 1994; Programmer, LDDS WorldCom, March 1995 to present.

Professional Memberships: Member of Sigma, Mu, Rho, the National Statistical Honor Society.