The

TI-89 Calculator

in the

Basic

Probability and Statistics

Course

LTC Mark B. Wroth

Department of Mathematical Sciences
United States Military Academy
West Point NY 10996

Revision 15.1

September 29, 2002

This is Revision 15.1, produced from ti89ma206.w on September 29, 2002.

The cartoon in Figure 2.1 is reproduced with the permission of the cartoonist, Cadet John K. Thompson, USMA 2004.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Why This Document?

This document was written as a supplement to assist students and teachers in MA206, the second year probability and statistics course required of all cadets at the United States Military Academy (USMA). This semester-long course is the final course in the core mathematics sequence. It is

> designed to advance [the student's] understanding of mathematical concepts and techniques used to model and analyze problems dealing with random effects and data[5].

Because of this focus, this course is not focused on the theory of probability and statistics; it is focused on the application of those sciences. Of course, it is not possible to fruitfully apply the tools of probability and statistics—or any other academic discipline—without a reasonable grounding in what is actually going on. But the focus of this course is on understanding of the tools applied to understanding of the real world.

Advanced calculators such as the HP-48 and TI-89 present both an opportunity and and a challenge to students of probability and statistics. On one hand, the calculator makes actually *performing* the sometimes tedious calculations needed in probability and statistics a matter of punching a few buttons. Advanced calculators also largely or completely eliminate the need for cumbersome tables. But this capability comes with a price. Not only does the student have to master the concepts of the course—a challenge in itself—but they must also learn what the capabilities of the calculator are and how to invoke them.

This document is aimed at students and teachers who are trying to master the aspects of the advanced calculator (specifically the TI-89) that apply to the basic probability and statistics course. It supplements the course textbook and the calculator handbook and focuses on those uses of the calculator specifically needed for this course. It covers both the built in operations of the calculator, and programs written specifically to assist with the subject.

We assume that the student has been using the same calculator through the core math sequence, and is therefor familiar with basic calculator operations. In addition to basic arithmatic computation, this includes symbolic manipulation, basic calculus (particularly numerical quadrature) and graphing of functions.

One of the powerful features of the advanced calculators is programmability. In addition to briefly covering the built in functions of the calculator, this document discusses some programs written to assist with subjects covered in the basic probability and statistics course, and provides some additional detail on the use of functions and programs that are part of the TI-89 and *TI-89 Statistics with List Editor flash application*[1]. The use of the calculator functions, and the implementation of the locally produced functions and programs, is discussed in Appendix C.

1.2 How to Use This Document

This document is generally divided into three main parts. Chapter 2 deals with topics that affect how the course is structured and overall strategies for using the calculator in a probability and statistics course. Chapters 4 through 9 discuss the major lesson blocks in MA206. Finally, the appendices provide additional details on specific topics.

Appendix A "Program and Function Reference" may be of special interest, as it gives use notes for the calculator programs defined in this document. It also gives the complete text of the programs (in ASCII form) and provides some discussion of selected programs from the *TI-89 Statistics with List Editor* application.

[1] A flash application is an additional software package loaded into the TI-89 to provide specialized additional capabilities. MA206 uses the *TI-89 Statistics with List Editor* flash application, available from the Texas Instruments' web site.

Chapter 2

General Issues in Learning Probability and Statistics Armed with an Advanced Calculator

2.1 Use of Graphing Calculators in Teaching and Learning Mathematics

It appears reasonable to believe—and there is some research evidence to suggest[1]—that how instructors model and require the use of the calculator strongly affects how students make use of the calculator. This is especially significant if the capabilities of the calculator to move between different representations of the data, or to make use of the Computer Algebra System (CAS) capabilities of the system, are to be used effectively.

This line of research also suggests that the way the instructor presents the process of thinking about mathematics in general, and probability and statistics in particular, affects how students use the calculator's capabilities (and how they view mathematics in general).

The implication of these thoughts is that instructors must become familiar with the tools and techniques they intend their students to take away from the course, and model them in the classroom. Since this requires some advanced preparation, both to decide how the calculator should be used and for the instructor to master the techniques involved. One of the

[1]See, for example, the research summary at [3]

functions of this document is to present both a point of view on this subject and to provide information allowing other instructors to reach their own conclusions.

The following sections are presented in an order generally conducive to a one semester course in probability and statistics, following the outline of MA206, the core course in the subject taught at USMA using [4] as the text. With some modification, it should be helpful in most basic probability and statistics courses.

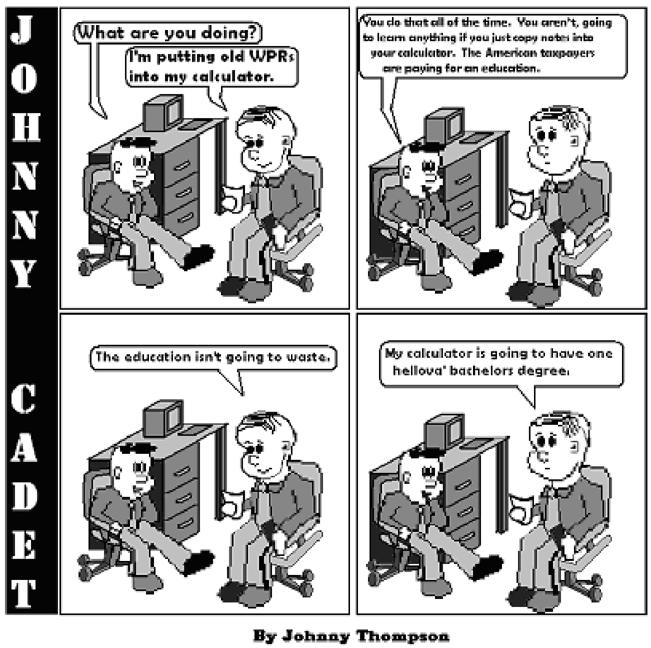

Figure 2.1: A cautionary note (with thanks to Cadet John Thompson, USMA '04)

The focus of many probability and statistics courses, and many students, appears to be on mastering the basic computations of the subject. For example, a major goal during a block on the exponential random variable is being able to correctly compute probabilities involving such a random variable. Facility with this calculation is then assumed later in the course. With a properly set up calculator, the calculation itself is simple; the challenge is in knowing when to use the distribution, what value to use for the parameter, and how to interpret the result.

The calculator can also largely replace the use of tables, and hence of the need to standardize random variables for most purposes. The exception to this is that many statistical packages, including the TI-89, use and display standardized random variables in statistical tests, so some understanding of the process is needed.

2.2 Interval Probabilities

Computing the probability that a random variable lies in a stated interval is a common task in the probability and statistics course. Especially with the capabilities of the TI-89, there are several valid strategies students may use for computing such probabilities:

- Manipulate the Probability Density Function (PDF) (or Probability Mass Function (PMF)) directly. For example. integrate the PDF over the interval. This approach implies the need for a user-manipulatable PDF (PMF) functions.

- Subtract the endpoint Cumlative Distribution Function (CDF) values. Most easily executed at the `Home` entry line, this approach implies the need for a user-manipulatable CDF for each distribution.

- Create (or find) a calculator program which computes the interval probability. Internally, such a program may use either computation approach.

There is no theoretical reason to choose between these techniques. Ideally, a student would master all of the different techniques and choose the technique appropriate to the particular problem.

Common practice in teaching the computation is to cover PDF-based approaches, but to emphasize CDF-based approaches. This fits well with the use of distribution tables, and may be easiest for some students because of this connection.

At the same time, the primary user interface for the probability computations in the TI-89 *TI-89 Statistics with List Editor* is a program based Graphical User Interface (GUI) which allows the user to enter the distribution parameters and the ends of the interval. While entry line functions are also provided in the *TI-89 Statistics with List Editor*, use of these functions from the entry line is essentially undocumented. Consistency with the general approach of the TI-89 would appear to suggest GUI-based interfaces are desirable. This requires writing programs for the distributions which are not included in the Statistics with List Editor application. For MA206, this would include the Hypergeometric, Uniform, and Exponential distributions.

The TI-89 allows the student to approach the calculation of interval probabilities any of the above ways—given the availability of either existing programs or basic programming skills for the third approach. To help gain understanding, it may be a good idea for students to focus on one method and ensure *it* is mastered. If GUI-based programs are available for all of the distributions of interest, focusing on this technique is likely to be the easiest.

2.3 Standardized Random Variables

The use of the calculator largely eliminates the need to use traditional probability tables. Since being able to use the standard normal probability tables is one of the main ways the use of a standardized random variable is presented, eliminating the need to use the tables at all also eliminates one of the major uses of standardized variables. It is tempting to simply ignore the topic completely if the student has adequate calculator skills.

However, there are several reasons to understand the basic manipulations surrounding standardized random variables, and the standard normal distribution in particular. Perhaps least important is the fact that traditional tables, while in some sense obsolescent as calculators with basic probability functions become more common, are still available when calculators are not, so some ability to use them is probably a good idea. More important from the perspective of the course material is that the manipulations to standardize the Normal random variable are the basis of the manipulations by which we derive the formulas for confidence intervals. So understanding how to standardize the normal random variable is a lead in to the material on confidence intervals. Finally, statistical packages—including the TI-89's advanced statistics functions— frequently state hypothesis test results in terms of standardized test statistics. Understanding the test results depends to some extent on understanding the normalized versions of the statistics.

Having said that, however, this document largely focuses on the direct manipulation of the random variables directly of interest in the problem. Standardized random variables are a special case of this focus.

Chapter 3

Calculator Tips

3.1 Split Screen

The calculator can split the main screen into two halves, displaying two independent applications. This is especially useful for working at the `Home` screen while keeping the *TI-89 Statistics with List Editor* application accessible.

To split the screen, go to the `Modes` menu, and set `Split 1 App` and `Split 2 App` to the two applications you want to use. Obvious candidates are `Home` and `Stats/List Editor`. Then set `Split Screen` to the mode you want; `Left Right` is probably the preferable option if the list editor is one of the applications.

This functionality is described in [9, Chapter14].

Chapter 4

Descriptive Statistics

4.1 Numerical Methods

The TI-89's `one variable statistics` application computes the sample mean, variance and standard deviation (using both the sample and population formulas), the median, and the quartiles of the sample.

4.1.1 Computing One-Variable Sample Statistics

1. Enter the sample data into a list.

2. Select the 1: 1-Var Stats option from the `[F4] Calc` menu (of the Statistics with List Editor application).

3. Enter the name of the list containing the sample data, either by entering the variable name directly, or by selecting `[2nd] VAR-LINK` and selecting the variable.

4. Select `ENTER` to confirm the selection, and again to compute the statistics.

   Tip: The one variable statistics output includes both `Sx` and σ`x`. Despite the similarity of the latter label in form to the symbol σ_x,the *population* standard deviation of the random variable X, it is not that quantity. Rather, it is

$$\sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n}} \tag{4.1}$$

While this statistic is sometimes useful, students tend to confuse it with the sample standard deviation

$$s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}} \tag{4.2}$$

which is almost always the quantity needed from a sample calculation. Put another way—if you need to *calculate* the standard deviation from sample data, you almost certainly want `Sx`, not σ`x`.

4.2 Graphical Methods

4.2.1 Histograms

A basic histogram is one of the standard plot types available. To create a histogram of data:

1. Enter the data into a list variable.
2. Select `[F2] Plots` to bring up the Plots menu.
3. Select `1:Plot Setup....`
4. Highlight a plot line.
5. Select `[F1] Define`.
6. Select `Plot Type`; on the popup menu, select `4:Histogram`
7. Enter the name of the list variable containing the data in the `x` box.
8. Enter an appropriate width for the histogram intervals in `Hist. Bucket Width`.

9. Ensure the `NO` option is selected in the `Use Freq and Categories?` popup.
10. Select `ENTER`.
11. Select `[F5] ZoomData`.

4.2.2 Box Plots

A basic box plot is one of the standard plot types available.

1. Enter the data into a list variable.
2. Select `[F2] Plots` to bring up the Plots menu.
3. Select `1:Plot Setup....`
4. Highlight a plot line.
5. Select `[F1] Define`.
6. Select `Plot Type`; on the popup menu, select `3:Box Plot`
7. Enter the name of the list variable containing the data in the `x` box.
8. Ensure the `NO` option is selected in the `Use Freq and Categories?` popup.
9. Select `ENTER`.
10. Select `[F5] ZoomData`.

The `[F3] Trace` function allows easy examination of the particular values included in the plot.

Chapter 5

Basic Operations

5.1 Counting

The TI-89 computes several basic functions useful for counting problems.

For most of these operations, there are two or three different ways to access the same calculator function:

- Select the function from a menu (usually the Math menu, accessed with the `[2nd MATH]` key).
- Type the name of the function in the Entry Line, using the alphabetic keys (or a menu pick from a custom menu).
- Add the function to the Entry Line using the `CATALOG`. Functions defined from flash applications (such as Statistics with List Editor) and user-defined functions are also available through the `CATALOG` function.
- Add the function to the entry line using the `Var Link` menu.

5.2 Counting Techniques

5.2.1 Factorials

The factorial function is accessed with the postfix operator `!` which can be entered from the keyboard (using the `[2nd] CHAR` function) or from the `[2nd] MATH 7: Probability` menu.

It is also found on the `Counting` menu of the MA206 custom menu.

5.2.2 Permutations

The "permutations" function can be accessed with the function `nPr(` function. This can be accessed via the `[2nd] MATH 7:  Probability nPr(` menu pick. It is also found on the `Counting` menu of the MA206 custom menu.

5.2.3 Combinations

The "combinations" function can be accessed through the `nCr(` function. This can be accessed via the `[2nd] MATH 7:  Probability nCr(` menu pick, or by typing the function name in the entry line. It is also found on the `Counting` menu of the MA206 custom menu.

Chapter 6

Random Variables and Probability Distributions

6.1 Discrete Random Variables

6.1.1 Arbitrary Random Variables

List operations used to examine and manipulate arbitrary discrete probability mass functions.

If the PMF can be described algebraically, or by a function defined in the TI-89, the summation operator can be used to compute quantities defined based on a summation over the entire PMF.

An example of such a quantity is the expected value. To compute the expected value of the function defined by `TIStat.binomPdf` (which happens to be the binomial distribution) using the definition

$$E(X) = \sum_{i=0}^{n} x\, p(x)$$

we could enter into the calculator

```
∑(x*TIStat.binomPdf(5,.2,x),x,0,5)
```

For this distribution, this is easily computed to be $np = 1$, confirming the result given by the calculator. Unfortunately, the calculator does not handle the general expression (that is, it will not compute the formula for the expected value in general).

For a more general arbitrary distribution, enter the values of the random variable into one list, and the associated probabilities into the corresponding positions of a second list. Entering the values from [4, page 110] into the lists named `x` and `p`, we can compute

```
∑(x[i]*p[i],i,0,7)
```

finding 4.57, the same result obtained by Devore. This notation parallels the traditional notation

$$\sum_{i=0}^{7} x_i\, p(i)$$

6.1.2 Binomial Distribution

The Binomial probability distribution is one of the pre-defined probability distribution in the *TI-89 Statistics with List Editor* application. It is accessed via the `[F5] Distr` menu, using either the `B: Binomial Pdf` or `C: Binomial Cdf` menu items.

It can also be accessed through the `binomGUI()` program described later in this document. Section A.4 describes the use and implementation of this function.

6.1.3 Hypergeometric Distribution

The Hypergeometric probability distribution is not one of the pre-defined distributions in the Statistics with List Editor. Since it is not pre-defined for us, we can define the PDF and CDF as TI-89 functions; `hypergeo` (Section A.5) is a program interface to those functions.

The PDF of the hypergeometric distribution is shown in

$$p_H(x) = \frac{\binom{Succ}{x}\binom{Pop - Succ}{n - x}}{\binom{Pop}{n}} \tag{6.1}$$

Where

`Pop` is the number of elements in the population, `Succ` is the number of elements coded "success",

`n` is the sample size,

and

$$\max(0, n - Pop + Succ) \leq x \leq \min(n, Succ) \tag{6.2}$$

The side conditions deal with the fact that the minimum number of successes in the sample is limited by the total number of failures in the population and the sample size (you can't have more failures in the sample than there are in the population), and the maximum number of successes in the sample is limited by the number of successes in the population.

6.1.4 Poisson Distribution

The Poisson probability distribution is one of the pre-defined probability distribution in the Statistics with List Editor application. It is accessed via the `[F5] Distr` menu, using either the `D: Poisson Pdf` or `E: Poisson Cdf` menu items.

6.2 Continuous Random Variables

6.2.1 Arbitrary Distributions

The TI-89's calculus applications can significantly ease the manipulation of arbitrary continuous probability distributions through their ability to find both definite and indefinite integrals. You reach these functions through the HOME screen, and should already be familiar from earlier calculus courses.

The major caution in applying the basic calculus functions to the PDF is to ensure that the limits of integration are correctly applied. Like any computer, the TI-89 will do what you tell it to, which may not be what you intended, particularly for piecewise defined functions!

Probabilities

Finding the probability that an arbitrarily defined continuous random variable lies in a given interval is, by definition, a matter of integrating the PDF over the interval. For simply defined functions (e.g. the Exponential distribution) this is easily accomplished with the Integrate function from the `HOME [F3] Calc` menu.

Expected Value

Finding the expected value of an arbitrarily defined continuous random variable can be accomplished by applying the definition of expected value. For simply defined functions (e.g. the Exponential distribution) this is easily accomplished with the Integrate function from the HOME **[F3] Calc** menu.

For example to find the expected value of

$$f(x) = \begin{cases} 0.15\,e^{-0.15(x-0.5)} & x \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

[4, Example 4.4], we compute

$$\int_{-\infty}^{0.5} 0\,dx + \int_{0.5}^{\infty} 0.15\,e^{-0.15(x-0.5)}\,dx$$

While this integral should be well within the capability of the student, it can also be solved as

```
∫(x*0.15*eˆ(-0.15*(x-0.5)),x,0.5,∞)
```

giving the result 7.16667.

> **Tip:** The TI-89 exponential operator, `eˆ(`, is not the letter `e`; it is entered using the `x` key.

Variance

Finding the variance of an arbitrarily defined continuous random variable can be accomplished by applying the definition of variance. For simply defined functions (e.g. the Exponential distribution) this is easily accomplished with the Integrate function from the HOME `[F3] Calc` menu. The computational formula,

$$V(X) = E(X^2) - [E(X)]^2$$

can be applied by integrating to find the expected value of X^2; this may not be easier than applying the definition directly.

Defining the PDF or the CDF as a TI-89 function allows it to be used in subsequent calculations. Examples of reasonable definitions are given for some of the probability distributions used in the basic probability and statistics course discussed below.

6.2.2 Uniform Distribution

The Uniform distribution is not a separately defined probability distribution in the Statistics with List Editor application. Therefore all manipulations of random variables with this distribution depend on manipulating the PDF directly, user defined programs, or on the use of known formulas.

The cumulative distribution function can be defined as a TI-89 function for convenience in calculation. An example of such a definition is shown in in the `unifcdf` reference page (Section A.8).

6.2.3 Exponential Distribution

The Exponential distribution is not a separately defined probability distribution in the Statistics with List Editor application. Therefore all manipulations of random variables with this distribution depend on manipulating the PDF directly, user-defined programs, or on the use of known formulas.

The `expcdf()` function (discussed in the Section A.9, `expcdf` reference page) can be used on the TI-89 to compute probabilities related to exponential random variables. For $\lambda > 0$, it implements the analytic probability distribution function:

$$f_X(x) = \begin{cases} \lambda\, e^{-\lambda x} & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

6.2.4 Normal (Gaussian) Distribution

The Normal, or Gaussian, probability distribution is one of the pre-defined distributions in the *TI-89 Statistics with List Editor* application. Because of its promenance in statistical applications, there are a variety of built-in functions for accessing and manipulating this distribution.

Computing Normal Probabilities

There are two main methods for computing probabilities involving the Normal distribution; the Normal Cdf function (accessed from the `[F5] Distr` menu of the *TI-89 Statistics with List Editor* application), and with the Shade function (also accessed from the `[F5] Distr` menu of the *TI-89 Statistics with List Editor* application). Both require the mean, standard deviation, and limits of the interval; the `Shade` function, in addition to computing the probability that the random variable is in the interval, draws the PDF and shades the area of interest.

Normal Probability Plots

The `[F2] Plots` menu includes the ability to create a normal probability plot of data in one of the lists. To draw a normal probability plot:

1. Start the *TI-89 Statistics with List Editor* application.
2. Enter the data into a list variable.
3. Select the Plots menu by pressing `[F2] Plots`.
4. Select `2:Norm Prob Plot`.
5. Fill out the resulting `Norm Prob Plot...` requestor:
   (a) Select an unused list variable at the `Plot Number` popup.
   (b) Enter the name of the list variable containing the data for which the probability plot is needed in the `List:` box.
   (c) Select values for the remaining entries on the requestor. The default values are probably acceptable.
6. Select `ENTER` to close the requestor.
7. Select the `Plots` menu by pressing `[F2] Plots`.
8. Select `1:Plot Setup`.

9. Select the plot variable containing the normal scores (the name of this variable was chosen in the `Plot Number` popup of the `Norm Prob Plot...`) by highlighting it using the cursor keys and pressing `[F4]`.
10. Display the plot by pressing `[F5] ZoomData` .

Chapter 7

Point Estimates

The TI-89 calculates a variety of sample statistics that can be used as point estimators for various quantities. The procedure for computing the most common of them is described above (in Section 4.1.1). This procedure produces a dialog box showing the following statistics:

$\bar{x}$ The sample mean.

$\sum x$ The sample total.

$\sum x^2$ The sum of the squared observations.

Sx The sample standard deviation, computed as

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

The "population" standard deviation, computed as

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

Note that this is the the true population standard deviation *only* if the *entire* population is part of the list on which the statistics are computed. *This value is not the population standard deviation for any other sample!* A common mistake made by students is to use this value for an unknown population standard deviation.

`n` The sample size.

`Min%` The sample minimum.

`Q1%` The first sample quartile; the value below which one quarter of the sample observations fall.

`Med%` The sample median; the value below which half of the sample observations fall.

`Q3%` The third sample quartile; the value below which three quarters of the sample observations fall.

`Max%` The sample maximum.

$\sum(x-\bar{x})^2$ Sum of the squared deviations.

The interval estimation functions provided by the TI-89 will also provide the point estimates of the relevant parameter when they compute the value based on a list containing the sample data.

Chapter 8

Confidence Intervals

> *The following development, more theoretical and expository than the rest of this document, is intended to provide an alternate introduction to confidence intervals more directly tied to the capabilities introduced by the advanced calculator than the equivalent material in [4].*

8.1 Purpose of Interval Estimates

The disadvantage of point estimates is that in isolation they provide no information about the accuracy of the estimate. Merely from a point estimate that, say, the average height of a randomly chosen student is 70 inches, the reader cannot tell if this value is precisely known, or only very approximately known.

Experimental scientists have been dealing with this issue for many years, and a variety of conventions for dealing with it have been tried. For example, one approach is to quote only the number of significant figures that are accurately known: 70 is different than 70.0 is different than 7×10^1. While this can be a convenient shorthand, it is counterintuitive in that from a mathematical sense all three of those representations express *exactly* the same number. It is also problematic in that there is no easy way to express the idea that "I'm pretty sure the value is between 68 and 72", an uncertainty that does not fall neatly into a power of ten.

This general concern leads to expressing a measurement as a point estimate plus or minus some value that expresses the uncertainty assigned by the experimenter to the value. This is the approach commonly taken by National Institute of Standards and Technology (NIST) in reporting its experimental results, and is probably familiar to the student from course work in the physical sciences. NIST's stated policy [8] is to express the *combined standard uncertainty* u_c, representing the estimated standard deviation of the measurement[1] This is commonly written as $\hat{\Theta} \pm u_c$ or, in our height example, 70 ± 2 inches. Another way to write this would be as the interval

$$\left(\hat{\Theta} - u_c, \hat{\Theta} + u_c\right)$$

where we are treating the value of the estimator as a random variable (because its value is not known in advance). How should we interpret this interval?

The answer lies in NIST's definition of u_c as the standard deviation of the measurement. An experimental measurement is a value computed from (usually) repeated observations of a quantity; in our terminology, a *sample* of all possible observations of that quantity. In other words, an experimental measurement is a *statistic*, and like any other random variable, it has a distribution and, therefore, a standard deviation[2] So the interval we are talking about is

$$\left(\hat{\Theta} - \sigma_{\hat{\Theta}}, \hat{\Theta} + \sigma_{\hat{\Theta}}\right)$$

Note

A truly rigorous discussion would at this point have to discuss the consequences of the fact that, while the experimenter would *like* to know $\sigma_{\hat{\Theta}}$, in reality he or she almost never does. Furthermore, the definition of u_c includes the experimenter's non-statistical estimates of the uncertainty of the measurement. Inclusion of these non-statistical estimates makes it difficult to proceed rigorously from the standpoint of mathematical statistics.

[1] The combined standard uncertainty includes estimates of both uncertainties estimated by statistical means (*Type A evaluation of uncertainty*) and those estimated by other means (*Type B evaluation of uncertainty*). Type A evaluation is generally what we are dealing with in a statistics course; Type B evaluation deals with the experimenter's evaluation of other sources of error, including experimental bias. An extended discussion of this is beyond the scope of this document; the curious ready may consult [8].

[2] In the most general case, it is not true that all random variables have finite standard deviation. But at this level of discussion, this refinement has little consequence.

In the development that follows, we will address some of the consequences of having to estimate $\sigma_{\hat{\Theta}}$. Dealing with the consequences of including non-statistical estimates of uncertainty is well beyond the scope of this paper.

If we know the probability distribution of $\hat{\Theta}$ and the value of the parameter θ, we could compute

$$\mathcal{P}\left(\hat{\Theta} - \sigma_{\hat{\Theta}} < \theta < \hat{\Theta} + \sigma_{\hat{\Theta}}\right)$$

the probability that the *random interval* contains the true value of the parameter. Notice that in this expression, unlike most of the probability statements we have encountered, the random variable is on the *ends* of the interval, *not in the center*. That is what the expression *random interval* means.

Further development of confidence intervals depends on what we know (or can assume) about the distribution of the population and the statistics we are collecting.

8.2 Samples from a Normal Population

In reality, of course, we do not know enough to make this computation. In fact, if we did, we would not need to; since the whole point of the exercise is to estimate θ, knowing its value would mean that we were done. But for some practical cases we can proceed anyway. Suppose we are sampling from a Normal population, and assume for the moment that we know σ^2 and are trying to estimate μ. The obvious estimator for μ is $\bar{X}$, since that statistic is the Minimum Variance Unbiased Estimator (MVUE) for μ. If the underlying population is Normally distributed, $\bar{X}$ is also Normally distributed, with $\mathcal{E}(\bar{X}) = \mu$ and $\mathcal{V}(\bar{X}) = \sigma^2/n$. Then

$$\mathcal{P}(\bar{X} - \sigma_{\bar{X}} < \mu < \bar{X} + \sigma_{\bar{X}})$$

is just the probability that a Normal random variable is within one standard deviation of its mean, or about 68%. We formally express this by saying that the interval $(\bar{X} - \sigma_{\bar{X}}, \bar{X} + \sigma_{\bar{X}})$ is a 68% *confidence interval* for μ. More generally, we would say that it is a $100(1 - \alpha)\%$ confidence interval, with $\alpha \approx 32\%$.

Conventionally, in order to allow us to choose probabilities other than 68%, we add a *coverage factor k* to the definition

$$\mathcal{P}\left(\hat{\Theta} - k\sigma_{\hat{\Theta}} < \theta < \hat{\Theta} + k\sigma_{\hat{\Theta}}\right)$$

$$\mathcal{P}\left(\bar{X} - k\sigma_{\bar{X}} < \mu < \bar{X} + k\sigma_{\bar{X}}\right)$$

By choosing the coverage factor k appropriately, we can make α any probability we want to. While the choice of $k = 1$ is suggested for the general reporting of experimental uncertainty by NIST other choices are conventional as well. Usually these other choices are expressed not by defining values of k, but by defining the resulting levels of either α (the *significance level*) or $100(1 - \alpha\%$ (the *confidence level*).

The expression can be manipulated to make understanding how to find k easier.

$$\mathcal{P}\left(\hat{\Theta} - k\sigma_{\hat{\Theta}} < \theta < \hat{\Theta} + k\sigma_{\hat{\Theta}}\right) = 1 - \alpha$$

$$\mathcal{P}\left(-k\sigma_{\hat{\Theta}} < \theta - \hat{\Theta} < +k\sigma_{\hat{\Theta}}\right) = 1 - \alpha$$

$$\mathcal{P}\left(-k < \frac{\theta - \hat{\Theta}}{\sigma_{\hat{\Theta}}} < +k\right) = 1 - \alpha$$

$$\mathcal{P}\left(k > -\frac{\theta - \hat{\Theta}}{\sigma_{\hat{\Theta}}} < -k\right) = 1 - \alpha$$

$$\mathcal{P}\left(k > \frac{\hat{\Theta} - \theta}{\sigma_{\hat{\Theta}}} > -k\right) = 1 - \alpha$$

$$\mathcal{P}\left(-k < \frac{\hat{\Theta} - \theta}{\sigma_{\hat{\Theta}}} < k\right) = 1 - \alpha$$

The numerical value of k depends on the distribution of the statistics involved in the confidence interval and on the desired probability α. The conventional notation in this case is $z_{\alpha/2}$; z because of the assumption that the distribution involved is Normal (because we are sampling from a Normal distribution, $\hat{\Theta} = \bar{X}$ is Normally distributed), and $\alpha/2$ because to put a probability of α outside the confidence interval, we need to put $\alpha/2$ outside each end of the interval.

More specifically, because we are dealing with distances away from the the mean of the distribution, and are scaling those distance by the standard deviation of the distribution, the k value in this case is based on a *standard*

Normal distribution, Z—one whose mean is zero and standard deviation is 1. To compute the numerical value, we need to compute the value $z_{\alpha/2}$ such that

$$\mathcal{P}(Z > z_{\alpha/2}) = \alpha/2$$

With the TI-89, this value is easy to find. Picking $\alpha = 0.1$, we can find $x_{0.05}$ using the inverse Normal function to be 1.64485.

Since the variance of $\bar{X}$ (given that we know σ) is σ^2/n. This confidence interval

$$\bar{X} \pm z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \tag{8.1}$$

gives a $100(1-\alpha)\%$ confidence interval for the mean of a Normal population whose standard deviation σ is known. This is the confidence interval that the TI-89 `ZInterval` function computes.

But we are working on the assumption that we know σ, the true population standard deviation. Usually this will not be true; if we know enough about the distribution to know its standard deviation, we also already know its mean! Because the situation—where we are dealing with a population modeled by a Normal distribution with unknown mean μ and standard deviation σ—is fairly common, considerable thought has gone into how to find the appropriate values. The approach starts with the random variable

$$T = \frac{\bar{X} - \mu}{\hat{\sigma_{\bar{X}}}} = \frac{\bar{X} - \mu}{\frac{S}{\sqrt{n}}}$$

This random variable has a *Student's T-distribution* with $n - 1$ degrees of freedom. The T-distribution is very similar to the standard Normal, Z, except that it has one parameter, the number of degrees of freedom ν. The effect of this parameter is to define how much more variable than the standard Normal the T is; the *smaller* ν is, the more variable the T distribution is. If $\nu = \infty$, the T distribution *is* the standard Normal. For finite values of ν, the variance of the T is greater than 1.

Reversing the process by which we explored the value of k, we can find that for this case the interval

$$\left(\bar{X} - k\frac{S}{\sqrt{n}}, \bar{X} - k\frac{S}{\sqrt{n}}\right)$$

gives an interval whose confidence level $100(1-\alpha)\%$ is determined by the coverage factor k in the same way as above. Using similar notation, we define $t_{\alpha/2,\nu=n-1}$ as the value that puts probability α/s outside each end of the interval, giving the interval

$$\left(\bar{X} - t_{t_{\alpha/2,n-1}} \frac{S}{\sqrt{n}}, \bar{X} - t_{\alpha/2,n-1} \frac{S}{\sqrt{n}}\right) \tag{8.2}$$

as a $100(1-\alpha/2)$ confidence interval. This is the confidence interval that the TI-89 calculates from the `TInterval` menu function.

Similarly, starting from the fact that if the population is Normal with mean μ and standard deviation σ the random variable

$$\frac{(n-1)S^2}{\sigma^2}$$

has a χ^2 distribution with $n-1$ degrees of freedom, we can find that the confidence interval

$$\left(\frac{(n-1)S^2}{\chi^2_{\alpha/2;n-1}}, \frac{(n-1)S^2}{\chi^2_{1-\alpha/2;n-1}}\right) \tag{8.3}$$

is a $100(1-\alpha)\%$ confidence interval for the variance of a Normal population. This is the confidence interval computed by the `Chi2int` and `Chi2GUI` TI-89 functions.

8.3 Large Samples from Non-Normal Populations

If the underlying population is not Normal, but has mean μ and finite variance σ^2, the distribution of $\bar{X}$ is *not* exactly known. However, if the size of the sample is large, the Central Limit Theorem argues that $\bar{X}$ is *approximately* Normal, with mean μ and variance σ^2/n. Under the same assumption that the sample size is large, it is reasonable to claim that S^2 is a very precise estimate of the population variance. If we assume this, we know the standard deviation of $\bar{X}$, and we can use the confidence interval defined by 8.1 (implemented by the TI-89 `ZInterval` function).

8.4 TI-89 Functions

The main TI-89 set of functions supporting interval estimation (other than the functions used for calculating sample statistics) is the `[F7] Ints` menu, which includes functions for Z and T-based confidence intervals on the mean (among others). These functions allow the interval to be calculated directly from sample data, or from previously computed sample statistics.

The `[F5] Distr` menu's `2: Inverse` submenu includes functions for computing the critical values of the Normal, Student's T, and χ^2 distributions.

> **Tip:** The various inverse functions ask for the `AREA` (probability) at which the inverse is to be calculated. This area is the probability that the random variable is *less than* the returned inverse value. This is consistent with the general definition of a CDF. However, the critical values of a distribution are defined in terms of the probability that the random variable is *greater than* the critical value. The translation between the two is, of course, that the area above the critical value is 1 minus the area the inverse function is expecting.

This difference can be ignored by taking the absolute value of the resulting critical value—if the distribution is symmetric around zero. Because this relationship does *not* hold true for distribution not symmetric about zero (i.e. the χ^2 distribution or the general normal distribution), relying on this property can lead the student into mistakes.

The `5: 1-PropZInt` menu can be used to calculate confidence intervals on the population proportion of a binomial distribution. However, this function appears to use the approximate formula defined by Devore [4, p. 291] in Equation 7.11 (which is the standard form used by most texts rather than the more exact form defined in Equation 7.10.

There do not appear to be direct functions for the calculation of χ^2 confidence intervals on variance. The `[F5] Distr` menu's `2: Inverse` submenu does include an `Inverse Chi-square` function which can be used to provide the Chi-squared critical values needed to compute the confidence intervals, however. The `chi2gui` calculator program is designed to fill this lack, and functions in a manner similar to the built-in TI-89 programs.

Chapter 9

Hypothesis Testing

The `[F6] Tests` menu includes applications for (among others), Z and T-based hypothesis tests. These tests allow the test statistic to be provided, or to be computed from data entered in one of the lists. In all cases, the calculator provides the *p*-value relevant to the test, rather than drawing a conclusion.

Appendix A

Program and Function Reference

This Appendix provides use information for selected TI-89 programs, including functions provided with the calculator, those belonging to the *Statistics with List Editor* application, and those provided in calculator programs defined in this document.

Implementation details are provided for functions and programs defined by this document. This includes a complete listing of the calculator code implementing them. See Appendix C for further details on the mechanics of this process.

A.1 General Notes on the Statistics with List Editor

The TIStat applications (belonging to the *TI-89 Statistics with List Editor* flash application) need their own reference pages; the application manual ([10]) does not describe how to use the applications from the Home screen, although they are available from the `CATALOG` screen as well as from the `MA206()` program. To some extent, this document attempts to fill that gap, at least with respect to the applications used in the basic probability and statistics course.

Students using the `TIStat.` applications should note that most of them store their outputs to the `statvars` directory.

A.2 "Features" of TI-89 Basic Programs

A.2.1 Auto-Alock

It is a sometimes annoying feature of the TI-89 that when the user is placed in a dialog box, the alpha lock is automatically turned on. While this behavior is reasonable if the actual data being requested is a string. But for many math programs (including most or all of the ones defined by this document) it means you must first press the α key to turn the α-lock back off.

If this behavior really irritates you, the "Auto Alpha-Lock Off" program by Kevin Kofler (available from `www.ticalc.org`) can reportedly prevent it.

A.3 MA206 Custom Menu

A.3.1 Name

`MA206()` — Set up a custom menu allowing easy access to functions commonly used in the basic probability and statistics course.

A.3.2 Description

This program sets up a TI-89 *custom menu*, which allows function names to be easily inserted into the Entry Line

> **Tip:** Function and program names can also be easily pasted into the Entry Line by using the `CATALOG` key. Once in the Catalog window, pressing `[F3] Flash Apps` will bring up a list of the functions installed by any flash applications, and `[F4] User-Defined` will bring up a list of user defined functions.

> **Tip:** When a function has been highlighted in either the `[F3] Flash Apps` or `[F2] Built-in` panes of the Catalog window, pressing `[F1] Help` will bring up a terse description of the inputs for the function.

A.3.3 Usage

Inputs

This program has no inputs.

Outputs

This program reconfigures the custom menu of the TI-89. It has no other outputs.

Example

A.3.4 TI-89 Implementation

This, and other programs defined by this document, are shown in the form exported by the `TI-89 Graph Link` program. Copies of the ASCII version of the program are found in the output directory, and can be imported into `Graph Link`.

"MA206.TXT" 40a ≡

```
\START92\
\COMMENT=
\NAME=MA206
\FILE=MA206.89P
()
Prgm
\(C)\ ARGS: none; Set up MA206 menu
\(C)\ Program to set up an MA206 custom menu
\(C)\ Rev 3.4 23 JUN 02
\(C)\ D/MathSci USMA (Mark Wroth)
Custom
◇
```

File defined by 40ab, 41ab, 42abc.

The `Tools` menu is used for functions that manipulate the calculator itself. The `MA206` entry reloads the custom menu (primarily for use in debugging).

"MA206.TXT" 40b ≡

```
Title "Tools"
 Item "NewProb"
 Item "MA206()"
 Item "CustmOff"
 Item "autoaoff()"
 Item "uninevhk()"
◇
```

File defined by 40ab, 41ab, 42abc.

The last two menu items are useful only if you have installed the `autoaoff` assembly program. See Section C.4.3 for discussion of this program. The `uninevhk` (for "uninstall event hook") function is part of the same package, and uninstalls `autoaoff`, restoring the default behavior of the calculator.

The `Calc` menu provides easy access to functions commonly needed in calculations.

`"MA206.TXT"` 41a ≡

```
Title "Calc"
 Item "\integral\("
 Item "\Sigma\("
 Item "\root\("
◇
```

File defined by 40ab, 41ab, 42abc.

The `Counting` menu provides access to several functions used specifically in counting problems. All of these functions are built in to the TI-89; this menu just makes accessing them convenient.

`"MA206.TXT"` 41b ≡

```
Title "Counting"
 Item "nPr("
 Item "nCr("
 Item "!"
◇
```

File defined by 40ab, 41ab, 42abc.

The `Distr` menu provides command line access to several functions related to computing probabilities directly from the command line.

"MA206.TXT" 42a ≡

```
Title "Distr"
 Item "ma206\binomGUI()"
 Item "TIStat.PoissPdf("
 Item "TIStat.PoissCdf("
 Item "ma206\hypergeo()"
 Item "ma206\expGUI()"
 Item "ma206\unifGUI()"
 Item "ma206\NormCDF()"
◇
```

File defined by 40ab, 41ab, 42abc.

The `Distr` menu provides command line access to functions related to computing confidence intervals.

"MA206.TXT" 42b ≡

```
Title "Intvl"
 Item "TIStat.zInt("
 Item "TIStat.tInt("
 Item "TIStat.zInt_1P("
 Item "BinomCI()"
 Item "Chi2GUI()"
◇
```

File defined by 40ab, 41ab, 42abc.

Finally, the `Functions` menu provides command line access to some of the functions underlying the other menus, specifically including the PDF and CDF functions.

"MA206.TXT" 42c ≡

```
Title "Functions"
 Item "TIStat.binomPdf("
 Item "TIStat.binomCdf("
 Item "ma206\hygeoPdf("
 Item "ma206\hygeoCdf("
 Item "ma206\expCdf("
 Item "ma206\unifCdf("
 Item "TIStat.normCdf("
 Item "Chi2Int("
```

```
EndCustm
CustmOn
EndPrgm

\STOP92\
```
◇

File defined by 40ab, 41ab, 42abc.

A.4 Binomial Distribution

A.4.1 Name

`binomGUI`, `TIStat.binomCdf` – These GUIs compute the probability that a binomial random variable lies in the specified (closed) interval. The `binomGUI` program provides access to the `binomCdf` program from the `Home` command line.

A.4.2 Usage

Inputs

`n` The number of trials

`p` The probability of success on each trial

`Lower Value` The lower endpoint of the interval over which the binomial probability is desired.

`Upper Value` The endpoint of the interval over which the binomial probability is desired.

The input dialog from `binomGUI` is shown in Figure A.1. The dialog box from the `Binomial CDF` application is similar.

Outputs

Example

A.4.3 TI-89 Implementation

"`BINOMGUI.TXT`" 43 ≡

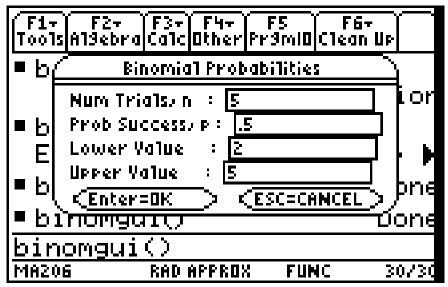

Figure A.1: `binomGUI` input screen

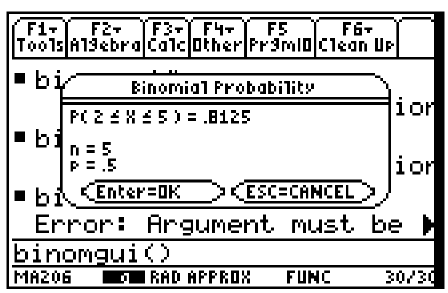

Figure A.2: `binomGUI` output screen

```
\START92\
\COMMENT=
\NAME=BinomGUI
\FILE=BINOMGUI.89P
()
Prgm
\(C)\ ARGS: none; Binomial probabilities
\(C)\ Compute interval probabilities
\(C)\ for a binomial distribution
\(C)\ USES: TIStat.binomCdf
\(C)\ Rev 1.2 21 JUN 02
\(C)\ D/Math Sci (Mark Wroth)
local n,p,lo,hi,cdf
Dialog
 Title "Binomial Probabilities"
 Request "Num Trials, n   ",n
 Request "Prob Success, p ",p
 Request "Lower Value     ",lo
 Request "Upper Value     ",hi
EndDlog
If ok=1 Then
 expr(lo)\->\lo
 expr(hi)\->\hi
 expr(n)\->\n
 expr(p)\->\p
 lo\->\statvars\LowVal
 hi\->\statvars\UpVal
 n\->\statvars\n
 p\->\statvars\p
 TIStat.binomCdf(n,p,lo,hi)\->\statvars\Cdf
 format(statvars\Cdf,"f5")\->\cdf
 string(statvars\LowVal)\->\lo
 string(statvars\UpVal)\->\hi
 string(statvars\n)\->\n
 string(statvars\p)\->\p
 Dialog
  Title "Binomial Probability"
  Text "P( "&lo&" \<=\ X \<=\ "&hi&" ) = "&cdf
  Text " "
  Text "n = "&n
```

```
  Text "p = "&p
 EndDlog
EndIf
EndPrgm
\STOP92\
◇
```

A.5 Hypergeometric Distribution

A.5.1 Name

hypergeo — Compute probabilities related to a hypergeometric distribution, specifically the probability that a hypergeometric random variable lies between two constants a and b, inclusive.

A.5.2 Usage

Description

`hypergeo` is a program which prompts the user for the parameters of a hypergeometric distribution and the endpoints of an interval, and then computes the probability that the random variable lies in that interval.

The hypergeometric distribution models a situation where a sample is taken from a finite population consisting of a fixed number of successes and failures *without* replacement. The random variable is the number of successes drawn in the sample.

The format of the program is intended to be similar to the format used in the Statistics with List Editor application.

A.5.3 Usage

The `hypergeo` program creates a requester that prompts the user for the necessary inputs and displays its outputs in another requester.

Inputs

The input requester for `hypergeo` is shown in Figure A.3.

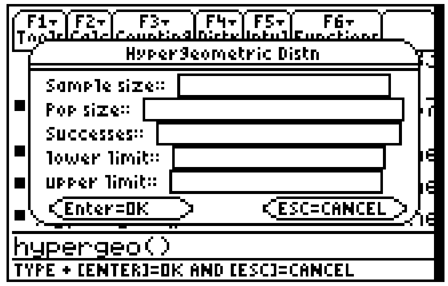

Figure A.3: `hypergeo` input screen

`Sample size` The size of the sample drawn.

`Pop size` The total size of the population from which the sample is drawn.

`Successes` The number of successes in the population.

`lower limit` The lower limit of the interval for which the probability is desired.

`upper limit` The upper limit of the interval for which the probability is desired.

Outputs

The primary output of the program is the probability that the random variable lies in the closed interval [a,b]. The program also echoes the parameters entered into the program as a check on data entry error. An example of the output requester that displays this information is shown in Figure A.4

A.5.4 Example

A.5.5 TI-89 Implementation

"hypergeo.txt" 47 ≡

```
\START92\
\COMMENT=
\NAME=hypergeo
```

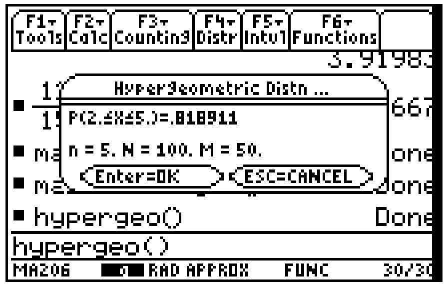

Figure A.4: hypergeo output screen

```
\FILE=HYPERGEO.89P
()
Prgm
\(C)\ ARGS: none; Hypergeometric prob (GUI)
\(C)\ Rev 2.3 21 JUN 02
\(C)\ D/MathSci USMA (Mark Wroth)
Local n,succ,pop,a,b,prob,usrmode
getMode("ALL")\->\usrMode
setMode("Exact/Approx","APPROXIMATE")
Dialog
 Title "Hypergeometric Distn"
 Request "Sample size:",n
 Request "Pop size:",pop
 Request "Successes:",succ
 Request "lower limit:",a
 Request "upper limit:",b
EndDlog
expr(n)\->\n
expr(pop)\->\pop
expr(succ)\->\succ
expr(a)\->\a
expr(b)\->\b
\(C)\ Check inputs
\(C)\ not implemented
\(C)\ Compute
If a<b Then
  \Sigma\(hygeoPdf(x,n,succ,pop),x,a,b)\->\prob
Else
```

```
  40\->\main\err
  PassErr
EndIf
\(C)\ Display probability
Dialog
 Title "Hypergeometric Distn ..."
 Text "P("&string(a)&"\<=\X\<=\"&string(b)&")="&string(prob)
 Text " "
 Text "n = "&string(n)&" N = "&string(pop)&" M = "&string(succ)
EndDlog
setMode(usrMode)
EndPrgm

\STOP92\
◇
```

A.6 hygeopdf

A.6.1 Name

`hygeopdf` — Evaluate the PMF of a hypergeometric random variable.

A.6.2 Description

`hygeopdf(x, n, succ, pop)` computes the probability that a hypergeometric random variable with sample size `n`, possible number of successes `succ`, and population size `pop` assumes the value `x`.

The hypergeometric PDF is defined in Equation 6.1.

A.6.3 Usage

Inputs

`x` The value at which the PDF is to be evaluated.

`n` The sample size.

`succ` The total number of successes in the population.

pop The total number of elements (successes and failures) in the population.

Outputs

probability The PDF value.

A.6.4 Example

To find the probability that a random variable from a hypergeometric distribution with a population size of 50 with 15 successes and a sample size of 10 has exactly 5 successes:

1. Enter hygeopdf(5.,10,15,50) in the entry line of the **Home** window.
2. Press `Enter`
3. The expression you entered and the answer, `.094903`, will be displayed in the History Area.

Note

If you enter all of the parameters using exact forms, the calculator will display the exact answer (in this case 904332/9529015). Entering any parameter using a decimal form (the 5. in the example) cause the calculator to provide the approximate answer.

A.6.5 TI-89 Implementation

Because of the very simple definition of hygeocdf(), it is important that we define hygeopdf() to return zero for invalid values of x. It is also appropriate to test for invalid parameter inputs; an invalid input here can propogate up to the CDF.

"hygeoPdf.txt" 50 ≡

```
\START92\
\COMMENT=
\NAME=hygeopdf
\FILE=HYGEOPDF.89P
hygeopdf(x,n,succ,pop)
```

```
Func
\(C)\ ARGS: (x,n,succ,pop) Hypergeometric PDF
\(C)\
\(C)\ Rev 2.0 20 JUN 02
\(C)\ D/MathSci USMA (Mark Wroth)
If n>pop or succ>pop Then
  Return "'Invalid parameters"
EndIf
If max(0,n-pop+succ)\<=\x and x\<=\ min(n,succ) Then
  nCr(succ,x)*nCr(pop-succ,n-x)/(nCr(pop,n))
Else
  0
EndIf
\(C)\ hypergeometric RV
\(C)\ Rev 1.1
\(C)\ Mark Wroth
EndFunc

\STOP92\
```
◇

A.7 hygeocdf

A.7.1 Name

`hygeocdf` — Evaluate the CDF of a hypergeometric random variable.

A.7.2 Description

`hygeocdf(x, n, succ, pop)` computes the probability that a hypergeometric random variable with sample size `n`, possible number of successes `succ`, and population size `pop` assumes a value less than or equal to `x`.

The hypergeometric PMF is defined in Equation 6.1; the CDF is

$$P_H(x) = \sum_{i=0}^{x} \frac{\begin{pmatrix} Succ \\ i \end{pmatrix}\begin{pmatrix} Pop - Succ \\ n - i \end{pmatrix}}{\begin{pmatrix} Pop \\ n \end{pmatrix}} \tag{A.1}$$

where

Pop is the number of elements in the population,

Succ is the number of elements coded "success",

n is the sample size.

Unlike the PDF, there are no limits (in principle) on x, although some care is needed to ensure that the function behaves properly at all values.

A.7.3 Usage

This function is called directly from the `Home` command line; the program `hypergeo` (Section A.5), also called from the command line, produces a requester to prompt the user for the necessary inputs.

Inputs

x The value at which the CDF is to be evaluated.

n The sample size.

succ The total number of successes in the population.

pop The total number of elements (successes and failures) in the population.

Outputs

probability The CDF value.

A.7.4 Example

To find the probability that a random variable from a hypergeometric distribution with a population size of 50 with 15 successes and a sample size of 10 has 5 or fewer successes:

1. Enter `hygeocdf(5.,10,15,50)` in the entry line of the `Home window`.
2. Press `Enter`

3. The expression you entered and the answer, `.969998`, will be displayed in the History Area.

Note

If you enter all of the parameters using exact forms, the calculator will display the exact answer (in this case 2813126/2900135). Entering any parameter using a decimal form (the 5. in the example) cause the calculator to provide the approximate answer.

A.7.5 TI-89 Implementation

The CDF for the Hypergeometric can be implemented easily given the existence of a PDF function which correctly returns zero for values of x which violate the side conditions (see ?).

"HYGEOCDF.TXT" 53 ≡

```
\START92\
\COMMENT=
\NAME=hygeoCdf
\FILE=HYGEOCDF.89P
hygeoCdf(x,n,succ,pop)
Func
\(C)\ ARGS: (x,n,succ,pop); CDF for a Hypergeometric RV
\(C)\ USES: hygeoPdf
\(C)\ Rev 1.1
\(C)\ Mark Wroth
 \Sigma\(hygeoPdf(i,m,succ,pop),i,0,x)
EndFunc

\STOP92\
◇
```

We depend on the error checking of `hygeopdf()` to catch any parameter errors, so the CDF does not need to do any independent error checking.

The `\(C)\` symbol represents the TI-89 comment symbol (see Appendix C).

A.8 unifcdf

A.8.1 Name

`unifcdf` — Evaluate the CDF of a uniformly distributed random variable.

A.8.2 Description

This function evaluates the CDF of a uniformly distributed random variable, defined as

$$F(x) = \begin{cases} 0 & x < 0 \\ \frac{x-a}{b-a} & a \le x \le b \\ 1 & x > b \end{cases} \tag{A.2}$$

A.8.3 Usage

Inputs

`x` The value at which the CDF is to be evaluated.

`a` The lower limit of the region for which the PDF is non-zero.

`b` The upper limit of the region for which the PDF is non-zero.

Outputs

cdf Cumulative probability, the probability that a uniformly distributed random variable with the specified parameters is less than or equal to x.

A.8.4 Example

To find the probabilitity that a random variable uniformly distributed between 1 and 10 is less than 5:

1. Enter `unifcdf(5.,1,10)` in the entry line of the `Home` window.
2. Press `Enter`

3. The expression you entered and the answer, `.444444`, will be displayed in the History Area.

Note

If you enter all of the parameters using exact forms, the calculator will display the exact answer (in this case 4/9). Entering any parameter using a decimal form (the 5. in the example) cause the calculator to provide the approximate answer.

A.8.5 TI-89 Implementation

"unifcdf.txt" 55 ≡

```
\START92\
\COMMENT=
\NAME=unifCdf
\FILE=UNIFCDF.89P
(x,a,b)
Func
\(C)\ ARGS: (x,a,b) CDF for a Uniform RV on [a,b]
\(C)\ Rev 1.1 2002-06-15
\(C)\ Mark Wroth
If a<b Then
 If x<a Then
  Return 0
 Elseif x>b Then
  Return 1
 Else
  (x-a)/(b-a)
 EndIf
Else
 "Parameters a < b"
EndIf
EndFunc

\STOP92\
◇
```

A.9 expcdf

A.9.1 Name

`expcdf` — Evaluate the CDF of an exponentially distributed random variable.

A.9.2 Description

This function implements the CDF for an exponentially distributed random variable with parameter λ. Such a random variable has PDF

$$f(x) = \lambda e^{-\lambda x} \tag{A.3}$$

A.9.3 Usage

Inputs

`x` The value of the random variable at which the CDF is to be evaluated.

λ The parameter of the distribution. λ is one over the mean of the distribution.

Outputs

`cdf` Cumulative probability; the probability that the random variable is less than or equal to the supplied x.

If an invalid parameter λ is supplied, an error string is returned, rather than a numeric result.

A.9.4 Example

To compute the probability that an exponentially distributed random variable with mean 5 is less than or equal to 3:

1. Enter `expcdf(3.,1/5)` in the entry line of the `Home` window.

2. Press `Enter`

3. The expression you entered and the answer, `.451188`, will be displayed in the History Area.

Note

If you enter both parameters using exact forms, the calculator will display the exact answer (in this case 1-e$^{-3/5}$). Entering either parameter using a decimal form (the 3. in the example) cause the calculator to provide the approximate answer.

A.9.5 TI-89 Implementation

The `expcdf()` function wraps a simple call to the usual mathematical definition inside two tests. The first of these tests checks that the required parameter λ is greater than zero, as required by the definition of the function. The second test checks whether the input value `x` is greater than or less than zero, branching to the two piecewise definitions of the CDF depending on the result. Both tests use the `where()` function, which is in essence a simple branching structure.

"EXPCDF.TXT" 57 ≡

```
\START92\
\COMMENT=
\NAME=expCdf
\FILE=EXPCDF.89P
(x,\lambda\)
Func
\(C)\ ARGS: (x,\lambda\) CDF of an exponential RV
\(C)\ with parameter \lambda\
\(C)\ Rev 1.1 2002-06-15
\(C)\ D/MathSci USMA (Mark Wroth)
when(\lambda\>0, when(x\>=\0,1-\e\^(\(-)\\lambda\*x),0),"\lambda\ must be > 0")
EndFunc

\STOP92\
```

◇

A.10 Exponential Probability GUI

A.10.1 Name

expgui—Produce confidence intervals on the variance or standard deviation of a Normal random variable.

A.10.2 Usage

A.10.3 TI-89 Implementation

"EXPGUI.TXT" 58 ≡

```
\START92\
\COMMENT=
\NAME=expGUI
\FILE=EXPGUI.89P
()
Prgm
\(C)\ ARGS: none; Exponential probabilities (GUI)
\(C)\ Rev 1.0 2002-06-15
\(C)\ D/MathSci USMA (Mark Wroth)
Local lo,hi,\lambda\,cdf,usrmode
getMode("ALL")\->\usrMode
setMode("Exact/Approx","APPROXIMATE")
Dialog
 Title "Exponential Distn"
 Request "Lower bound",lo
 Request "Upper bound",hi
 Request "\lambda\          ",\lambda\
EndDlog
If ok=1 Then
 expr(lo)\->\lo
 expr(hi)\->\hi
 expr(\lambda\)\->\\lambda\
 (ExpCDF(hi,\lambda\)-ExpCDF(lo,\lambda\))\->\cdf
 Dialog
  Title "Exponential Distn ..."
  Text "P("&string(lo)&" \<=\ X \<=\ "&string(hi)&") = "&format(cdf,"f4")
  Text " "
```

```
  Text "\lambda\ = "&string(\lambda\)
 EndDlog
 setMode(usrMode)
EndIf
EndPrgm

\STOP92\
◇
```

A.11 Confidence Intervals on the Binomial Proportion

A.11.1 Name

`TIStat.zInt_1P(x, n, clevel` — compute confidence interval on the population proportion for a binomial distribution.

`BinomCI()` — Calculate and store a confidence interval on the population proportion for a binomial distribution using [4, Equation 7.10].

A.11.2 Discussion

There are two different formulas for a confidence interval on a population proportion given by Devore in equations 7.10 and 7.11. The TI-89 computes a confidence interval for the population, but the documentation does not specify which formula it uses.

The two formulæ are:

$$\frac{\hat{p} + \frac{z_{\alpha/2}^2}{2n} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n} + \frac{z_{\alpha/2}^2}{4n^2}}}{1 + (z_{\alpha/2}^2)/n} \tag{A.4}$$

valid for all n[1] and

$$\hat{p} \pm z_{\alpha/2}^2 \sqrt{\hat{p}\hat{q}/n} \tag{A.5}$$

valid when $n\hat{p} \geq 10$ and $n\hat{q} \geq 10$.

[1] Devore, citing [1], indicates that this expression gives a more accuract confidence interval than Equation A.5 and the side conditions need not be checked for this form.

To determine which equation the TI-89 uses, we take a simple test case. Since the approximation given by Equation A.5 will improve with larger n, we will choose as our test case $n = 2$ and number of successes 1. With this assumption, $\hat{p} = \hat{q} = 0.5$. Selecting $\alpha = 0.5$ (and hence $z_{\alpha/2} = 0.67449$), we have now determined the values we need to use for our calculations.

Using Matlab as a calculator, the lower limit calculated with Equation A.4 is 0.28475881963799. The equivalent value for Equation eqn:7.11 is 0.26153177357874.

Applying the 1 Proportion Z Interval function of the TI-89 with $Success = 1$, $n = 2$, and $CI = 0.5$, the calculator returns a confidence interval of $(0.2615, 0.7385)$. Since Equation A.4 would have returned (at that level of precision) 0.2753 and Equation A.5 would have returned 0.2615, we can conclude that the calculator uses Equation A.5 or something very similar to it.

A.11.3 Usage

There are three different ways to access this basic functionality:

- Select the *TI-89 Statistics with List Editor* `F7 Ints` menu item `5: 1-PropZInt` and fill in the required values. Outputs are displayed in a dialog box and stored in the `StatVars` directory.
- Enter at the command line `TIStat.Zint_1P(x,n,clev)`. Outputs are stored in the `StatVars` directory, but not displayed.
- Enter at the command line `ma206/binomCI()` and fill in the required values in the resulting dialog box. Outputs are displayed in a dialog box and stored in the `StatVars` directory.

Both all three methods take (by design) essentially the same inputs and provide the same outputs; they are adapted to different user interfaces and slightly different underlying equations (as discussed above). In general, use `binomCI` is slightly preferable; it is definitely preferred when n is not “large”.

Inputs

`x` Number of observed successes

n Number of trials

clev The desired confidence level.

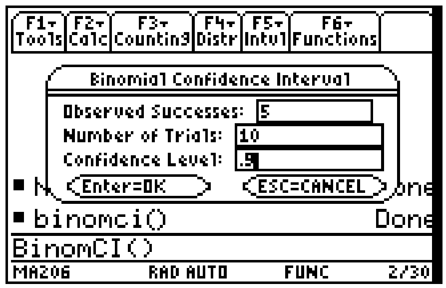

Figure A.5: `BinonCI` input screen

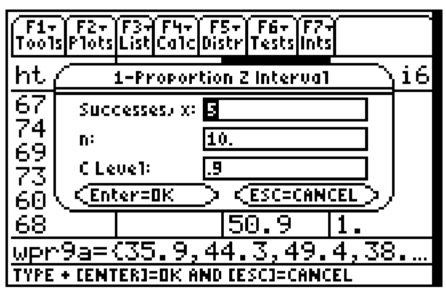

Figure A.6: `1-PropZInt` input screen

Outputs

statvars\lower Lower end of the CI.

statvars\upper Upper end of the CI.

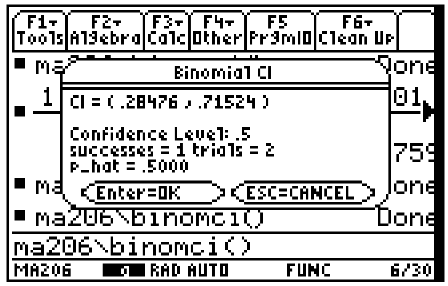

Figure A.7: `BinonCI` output screen

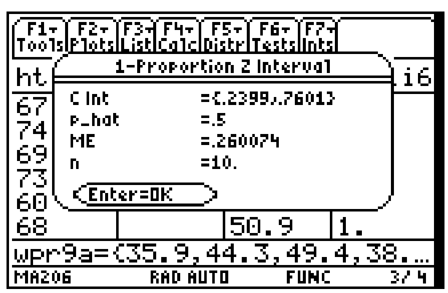

Figure A.8: `1-PropZInt` output screen

Example

To compute a 90% confidence interval on the population proportion, given 5 successes out of 10 trials, enter

`Observed Successes` 5

`Number of Trials` 10

`Confidence Level` .9

as shown in Figure A.5 (for the `binomCI` program) or Figure A.6 (for the `1-PropZint` program).

The results, a CI of $(0.28476, 0.71524)$ from `binomCI` (shown in Figure A.7), or $(0.2615, 0.7385)$ from `1-PropZInt` (shown in Figure A.8). As discussed above, the difference in the outputs is due to the implementing equations and the very small sample size chosen for the example.

A.11.4 TI-89 Implementation (binomCI)

"BINOMCI.TXT" 63 ≡

```
\START92\
\COMMENT=
\NAME=BinomCI
\FILE=BINOMCI.89P
()
Prgm
\(C)\ ARGS: none; CI on a binomial proportion (GUI)
\(C)\ Rev 2.0 2002-06-18
\(C)\ D/MathSci USMA (Mark Wroth)
Local x,n,clev,p_hat,z,t1,t2,t3,lower,upper
string(statvars\x)\->\x
string(statvars\n)\->\n
string(statvars\clevel)\->\clev
Dialog
 Title "Binomial Confidence Interval"
 Request "Observed Successes",x
 Request "Number of Trials",n
 Request "Confidence Level",clev
EndDlog
```

```
If ok=1 Then
 expr(x)\->\statvars\x
 expr(n)\->\statvars\n
 expr(clev)\->\statvars\clevel
 expr(x)\->\x
 expr(n)\->\n
 expr(clev)\->\clev
 x/n\->\p_hat
 p_hat\->\statvars\p_hat
 invNorm(1-(clev/2))\->\z
 z*\root\((p_hat)*(1-p_hat)/n+z^2/(4*n^2))\->\t1
 (1+z^2/n)\->\t2
 p_hat+z^2/(2*n)\->\t3
 (t3-t1)/t2\->\statvars\lower
 (t3+t1)/t2\->\statvars\upper
 format(statvars\lower,"f5")\->\lower
 format(statvars\upper,"f5")\->\upper
 Dialog
  Title "Binomial CI"
  Text "CI = ( "&lower&" , "&upper&" )"
  Text " "
  Text "Confidence Level: "&string(statvars\clevel)
  Text "successes = "&string(x)&" trials = "&string(n)
  Text "p_hat = "&format(Statvars\p_hat,"f4")
 EndDlog
EndIf
EndPrgm


\STOP92\
◇
```

A.12 Confidence Intervals on Variance—chi2int

A.12.1 Name

chi2int — The chi2int() program (and its companion chi2gui(), which provides a graphical user interface to the program) computes confidence intervals on the population variance or standard deviation.

A.12.2 Usage

Inputs

This function can be called in either of two ways: from the Home command line, as `chi2int(n,s2,clevel,type)` or by calling `chi2gui()`. If the `chi2int` for is used, the input arguments are:

`n` The number of samples in the sample.

`s2` The sample variance.

`clevel` The desired confidence level for the confidence interval.

`type` The type of interval desired, where `1` indicates a confidence interval on the variance, and `2` a confidence interval on the standard deviation.

If the `chi2gui()` form is used, there are no command line inputs; the program will raise a requester to allow the user to supply the needed values.

Outputs

The `chi2int` program provides its outputs in two forms: a graphical requester that provides the requested confidence interval and echoes the user inputs, and by storing the user inputs and the desired confidence interval endpoints in the `statvars` directory.

The set of stored variables are different for the `chi2int` and the `chi2gui` programs. The chi2int stores:

`statvars\lower` The lower end of the desired confidence interval.

`statvars\upper` The upper end of the desired confidence interval.

In addition, the `chi2gui` will store the following user inputs to the indicated variables (and will use the values in those variables as the default choices when it opens).

`statvars\n` The sample size

`statvars\ssdevx` The sample standard deviation (square root of the entered sample variance.

`statvars\clevel` The confidence level.

A.12.3 Example

Given a sample of size n = 17, and a sample variance of 137,324.3, compute a 95% confidence interval on the population variance.

1. Begin at the Home screen.

2. Enter the command `chi2int(17,137324.3,.95,1)` and press `Enter`.

    Tip: As a shortcut to entering the command name, use the Catalog function and select the F4 User-Defined tab. Then select the desired function from the list.

3. Read the confidence interval (76171.3, 318080) on the resulting requester.

Alternatively, using the `chi2gui` to solve the same problem:

1. Start the `chi2gui` by entering `chi2gui()` at the `Home` screen.

2. Enter the values for n, the sample variance, and the confidence level in the open requesters.

3. Select the desired confidence interval type from the drop down menu.

4. Press `Enter`.

5. Read the confidence interval (76171.3, 318080) on the resulting requester.

Note

This is Example 7.15 from [4]. However, the use of the full accuracy of the Chi-squared inverse function rather than the five significant figures available from a set of tables results in a slightly different answer than Devore obtains.

A.12.4 TI-89 Implementation

"chi2int.txt" 67 ≡

```
\START92\
\COMMENT=
\NAME=chi2int
\FILE=CHI2INT.89P
chi2int(n,s2,clevel,type)
Prgm
\(C)\ ARGS: (n,s\2\,clevel,type) Chi\2\ CI (type 1 on var, 2 on SD)
\(C)\ Rev 1.2 2002-06-15
\(C)\ D/MathSci USMA (Mark Wroth)
local l,u,stsr
(n-1)*s2/(tistat.invchi2(1-(1-clevel)/2,n-1))\->\l
(n-1)*s2/(tistat.invchi2((1-clevel)/2,n-1))\->\u
"CI on \sigma\\^2\"\->\tstr
If type=2 Then
 \root\(l)\->\l
 \root\(u)\->\u
 "CI on \sigma\"\->\tstr
EndIf
Dialog
 Title tstr
 Text "Cint = ( "&string(l)&" , "&string(u)&" )"
 Text "n = "&string(n)&" S\^2\ = "&string(s2)
EndDlog
l\->\statvars\lower
u\->\statvars\upper
\root\(s2)\->\statvars\ssdevx
clevel\->\statvars\clevel
EndPrgm
\STOP92\
◇
```

A.13 Normal CDF

A.13.1 Name

normCDF is a user-defined program that partially duplicates the function of TIStat.normCdf. The primary use for this function is to be pasted into the

Home screen command line and provide a GUI for data entry and results.

A.13.2 Usage

A.13.3 TI-89 Implementation

"NORMCDF.TXT" 68 ≡

```
\START92\
\COMMENT=
\NAME=NormCDF
\FILE=NORMCDF.89P
()
Prgm
\(C)\ ARGS: none; Normal probs (GUI)
\(C)\ Compute normal probabilities
\(C)\ using the Gaussian (Normal) distn
\(C)\ REQUIRES: TIStat.NormCDF
\(C)\ Rev 1.0 2002-06-15
\(C)\ D/MathSci USMA (Mark Wroth)
Local lo,hi,\mu\,\sigma\,cdf
getMode("ALL")\->\usrMode
setMode("Exact/Approx","APPROXIMATE")
Dialog
 Title "Normal CDF"
 Request "Lower Value ",lo
 Request "Upper Value ",hi
 Request "\mu\           ",\mu\
 Request "\sigma\           ",\sigma\
EndDlog
If ok=1 Then
 expr(lo)\->\lo
 expr(hi)\->\hi
 expr(\mu\)\->\\mu\
 expr(\sigma\)\->\\sigma\
 TIStat.normCDF(lo,hi,\mu\,\sigma\)\->\cdf
 lo\->\statvars\lo
 hi\->\statvars\hi
 \mu\\->\statvars\\mu\
 \sigma\\->\statvars\\sigma\
 cdf\->\statvars\cdf
```

```
 Dialog
  Title "Normal CDF"
  Text "CDF         = "&string(cdf)
  Text " "
  Text "Lower Value = "&string(lo)
  Text "Upper Value = "&string(hi)
  Text "\mu\           = "&string(\mu\)
  Text "\sigma\           = "&string(\sigma\)
 EndDlog
EndIf
setMode(usrMode)
EndPrgm

\STOP92\
```
◇

Appendix B

Installation

This document assumes that you have successfully installed the *TI-89 Statistics with List Editor* application. This may also have required you to install a newer version of the operating system itself. If you need help with these procedures, see Appendix E.

Depending on what you have available, there are several possible approaches to the problem of installing the programs this document creates. The obvious ones are:

- Copy them from another TI-89 that already has them installed. This is probably the easiest method—provided you have access to another calculator that already has them. Copying the programs is discussed (briefly) in Section B.1 and in the calculator documentation.

- Upload them from a personal computer. This is fairly straightforward, provided you have a computer and the necessary connecting cable. This approach is discussed in Sections B.2 (entering programs via *TI-Graph Link*, B.3 (importing programs from this document), and B.4 (actually uploading the programs).

- Type them directly into the calculator. This approach is always feasible, but is also the most painful since the TI-89 is not exactly set up for the rapid entry of text. If you need to take this approach, you will need to read at least parts of Section C.2, which discusses the format in which the programs are shown in this document.

B.1 Copying Programs from Another TI-89

B.2 Entering Programs in TI-Graph Link

Typing the programs in this document into *TI-Graph Link* is fairly straightforard if you are already familiar with programming the TI-89. You will need to review Section C.2 to understand the conventions used in the ASCII format programs.

B.3 Extracting Programs from the Source of this Document

To extract the programs from the source of this document, `ti89ma206.w`

1. Process the source of this document, `ti89ma206.w`, with `Nuweb` by commanding

   ```
   nuweb ti89ma206.w
   ```

   This will produce, in addition to several other intermediate files, a series of text files such as `MA206.TXT`. The a list of the output files produced is in Section I.1 (note that it includes files such as `calculator.sty` that you would *not* want to upload).

B.4 Uploading Programs with TI-Graph Link

Before beginning upload of the programs to the calculator, create a folder on the calculator in which the programs will be stored. This document, and the `MA206` custom menu, assume that the programs will be stored in `ma206`. Make this folder current on the TI-89.

1. Start the *TI-Graph Link* software.
2. Load the program into *TI-Graph Link*

(a) Import the text file into `Graph Link` by selecting `Tools`, then `Import: ASCII Program ...`. Select the appropriate text file in the requester that appears, and click "OK". This will import the program into `Graph Link`.

(b) Load the `.89p` file into *TI-Graph Link* by selecting `File`, then `Open` and selecting the desired file from the resulting requester.

3. Send the program to the calculator using `Graph Link` (following the instructions in the `Graph Link` user manual, [12, page 7ff]).

If you have access to the programs in the from of a TI-89 *program group*, usually named `MA206PRO.89g`, they can be uploaded as a single entity. Otherwise you will have to import each of them individually and send them to the calculator.

Appendix C

Programming Notes

C.1 Literate Programming

Literate programming is a method of creating a computer program such that both the documentation of that program and the actual source code can be presented to human readers in a manner intended to make the program as easy to understand as possible, and correctly record the actual program as presented to the computer itself.

This document is a *literate program* in the sense that the program listings found in this document are copied to disk as part of processing this document (with *Nuweb* [2]). The resulting text files can then be imported into `Graph Link` and ultimately downloaded to the calculator. In this manner, the version of the program displayed here is kept current with the version actually loaded on the calculator.

Using this system is discussed in Appendix B (with regard to creating the actual calculator files) and in Appendix F (with regard to producing the documentation).

C.2 ASCII Format for TI-89 Programs

Because of the desire to record the programs in this document, and the literate programming orientation of it, the calculator programs shown here are in the (**ASCII!**) export format of the TI-GraphLink software.

The format of an ASCII text file for a TI-89 program is, as far as I can see, undocumented. This makes editing the programs something of a challenge.

In general, while I use the ASCII storage form for inclusion in this document, most program development is done in the `Graph Link` editor. However, some observations can be made about the ASCII export format.

- The backslash character, "\", appears to serve as an escape character, preceding and following special characters, and preceding certain keywords at the beginning of the file.
- Certain specific characters appear to be as shown in Table C.1.
- Note that some symbols are not the same as the obvious key symbol. This specificially includes the exponential symbol and the negation operator (which is not the same as the subtraction operator).

ASCII	Calculator	Meaning
`\(C)\`	©	Comment
`\lambda\`	λ	Greek letter
`\mu\`	μ	Greek letter
`\sigma\`	σ	Greek letter
`\->\`	$\rightarrow$	Store command
`\Sigma\`	$\sum$	Summation command
`\root\`	$\sqrt{}$	Square root command
`\e\`	e	Exponential
`\(-)\`	$-$	Negation
`\<=\`	$\leq$	Math symbol
`\>=\`	$\geq$	Math symbol

Table C.1: Control sequences in ASCII export files

Tables C.2 through C.6 are an expanded version of the Table C.1 based on [13].

The general format of the ASCII program files is a header of the form

```
\START92\
\COMMENT=
\NAME=NormCDF
\FILE=NORMCDF.89P
```

where the strings `NormCDF` and —NORMCDF.89P— are replaced by the appropriate names.

This is followed by the argument list in parentheses, following which is either a `Prgm EndPrgm` block or a `Func EndFunc` block. These blocks contain the appropriate program code.

Finally the program ends with a trailer of the form

```
\STOP92\
```

`\alpha\`	α	alpha	`\beta\`	β	beta
`\Gamma\`	Γ	capital gamma	`\gamma\`	γ	gamma
`\Delta\`	Δ	capital delta	`\delta\`	δ	delta
`\epsilon\`	ϵ	epsilon	`\zeta\`	ζ	zeta
`\theta\`	θ	theta	`\lambda\`	λ	lambda
`\xi\`	ξ	xi	`\Pi\`	Π	capital pi
`\pi\`	π	pi	`\mu\`	μ	mu
`\rho\`	ρ	rho	`\Sigma\`	Σ	capital sigma
`\sigma\`	σ	sigma	`\tau\`	τ	tau
`\phi\`	ϕ	phi	`\psi\`	ψ	psi
`\Omega\`	Ω	capital omega	`\omega\`	ω	omega

Table C.2: Greek letters in ASCII export files

C.3 Toolbars

The `Toolbar` block can be used to replace the standard menu with a user defined set of menus that can actually execute blocks of code. This has real potential for improving the user interface. However, it is not clear how to take advantage of this.

The principal issue is that while the custom toolbar is active, it supercedes all of the other calculator functions—including the home screen.

The following code illustrates the use of this functionality.

"MA206CMD.TXT" 75 ≡

```
\START92\
\COMMENT=
```

`\->\`	$\rightarrow$	'STORE'	`\option\`	◇	Option
`\union\`	$\cup$	Set union	`\intersect\`	$\cap$	Set intersect
`\subset\`	$\subset$	Set subset	`\element\`	$\in$	Set element
`\ee\`	E	EE key (exponent)	`\e\`	e	italic exponential 'e'
`\i\`	i	imaginary number 'i'	`\r\`		radian conversion 'r'
`\t\`		matrix transposition 't'	`\xmean\`		stat mean of x
`\ymean\`		stat mean of y	`\<=\`	$\leq$	less than or equal
`\!=\`	$\neq$	not equal to	`\>=\`	$\geq$	greater than or equal
`\/_\`	$\angle$	angle character	`\diff\`	d	differential d
`\integral\`	$\int$	integration	`\infinity\`	∞	infinity symbol
`\root\`	$\sqrt{}$	radical	`\(C)\`	©	Comment symbol
`\(-)\`	$-$	unary minus	`\o\`	$^\circ$	degrees symbol

Table C.3: Mathematical and programming operators in ASCII export files

```
\NAME=MA206CMD
\FILE=MA206CMD.89P
()
Prgm
\(C)\ Experimental program to execute commands
\(C)\ directly
Lbl start
ToolBar
 Title "Tools"
  Item "Exit", quit
  Item "Clear", clear
 Title "Distr"
  Item "Binomial", binomial
  Item "Hypergeo", hyg
  Item "Uniform", unif
  Item "Exponential", exp
  Item "Normal", nml
EndTbar
Lbl binomial
 ma206\binomGUI()
Goto start
Lbl hyg
 ma206\hypergeo()
```

\lock\		The LOCK character
\check\		The CHECK mark char
\block\		The centered block char
\from\		block right arrow
\to\		block left arrow
\up\		block up arrow
\down\		block down arrow
\leftarrow\		left arrow
\uparrow\		up arrow character
\downarrow\		down arrow character
\left\		big block left arrow
\right\		big block right arrow
\shift\		shift up arrow character
\...\		ellipsis (...)
\cent\		cent character
\pound\		pound sterling
\starburst\		starburst character
\yen\		yen character
\split\		split vertical bar character
\section\		section character
\a_\		feminine ord character
\<<\		double left arrow character
\lnot\		logical not character
\(R)\		registered mark character
\^-\		superscript minus character
\^+\		superscript plus character
\^2\		superscript 2 (square)
\^3\		superscript 3 (cube)
\^-1\		superscript -1 (inverse)
\para\		paragraph symbol
\.\		dot mark
\^x\		superscript x character
\^1\		superscript 1 character
\o_\		masculine ord character
\>>\		double right arrow

Table C.4: Special characters in ASCII export files

\ud!\		upside down exclamation
\ud?\		upside down question mark
\A`\		capital A grave
\A'\		capital A acute
\A^\		capital A circumflex
\A~\		capital A tilde
\A..\		capital A dieresis
\Ao\		capital A ring
\AE\		capital AE
\C,\		capital C cedilla
\E`\		capital E grave
\E'\		capital E acute
\E^\		capital E circumflex
\E..\		capital E dieresis
\I`\		capital I grave
\I'\		capital I acute
\I^\		capital I circumflex
\I..\		capital I dieresis
\-D\		capital D bar
\N~\		capital N tilde
\O`\		capital O grave
\O'\		capital O acute
\O^\		capital O circumflex
\O~\		capital O tilde
\O..\		capital O dieresis
\x\		times 'x' mark
\O/\		capital O slash
\U`\		capital U grave
\U'\		capital U acute
\U^\		capital U circumflex
\U..\		capital U dieresis
\Y'\		capital Y acute
\I>\		capital I (p)

Table C.5: Upper case international letters in ASCII export files

\ss\		eschett
\a‘\		a grave
\a’\		a acute
\aˆ\		a circumflex
\a˜\		a tilde
\a..\		a dieresis
\ao\		a ring
\ae\		ae
\c,\		c cedilla
\e‘\		e grave
\e’\		e acute
\eˆ\		e circumflex
\e..\		e dieresis
\i‘\		i grave
\i’\		i acute
\iˆ\		i circumflex
\i..\		i dieresis
\-d\		d-bar
\n˜\		n tilde
\o‘\		o grave
\o’\		o acute
\oˆ\		o circumflex
\o˜\		o tilde
\o..\		o dieresis
\/\		division symbol
\o/\		o slash
\u‘\		u grave
\u’\		u acute
\uˆ\		u circumflex
\u..\		u dieresis
\y’\		y acute
\i>\		ip
\y..\		y dieresis

Table C.6: Lower case international letters in ASCII export files

```
Goto start
Lbl unif
 ma206\unifGUI()
Goto start
Lbl exp
 ma206\expGUI()
Goto start
Lbl nml
 ma206\normCDF()
Goto start
Lbl clear
 ClrHome
Goto start
Lbl quit
EndPrgm

\STOP92\
```
◇

C.4 Random Notes

C.4.1 Help Strings

The `catalog` screen reports as "help" information the first comment line after the `Prgm` or `Func` line of the program. This line can be longer than an output line; it will be wrapped into a dialog box when the user presses `F1`. However, only the first few characters will be displayed at the bottom of the catalog screen.

C.4.2 Unavailable Functionality

It does not appear to be possible to insert a function prototype (i.e. to give variable names for arguments to a function to be pasted into the entry line) using a custom menu.

It does not appear possible to allow a variable number of parameters to be passed to a function or program, or to branch within a function or program based on the number of supplied parameters.

C.4.3 Automatic Alpha Lock

The TI-89 automatically turns on the "alpha lock" when a dialog box is created using the `Request` command. While this would be convenient if strings were the expected input, it is less so in these applications. However, I do not know a way to turn the alpha lock back off or to keep it from turning on. the "Auto Alpha-Lock Off" program by Kevin Kofler (available from `www.ticalc.org`) can prevent this behavior.

C.4.4 Additional Programming References

References [6] and [7] provide some additional insight into programming the TI-89.

Appendix D

Revision History

This Appendix gives the revision history of this document as a whole.
Revision 15.0*(2002/09/29 18:50:44; MarkWroth)* Recovering the master files from backup following a computer crash. Adding and clarifying references to literate programming, including specifically comments on the use of *Nuweb*.
Revision 14.10*(2002/06/23 22:21:10; WrothMark)* Fixed a command typo.
Revision 14.9*(2002/06/23 22:18:13; WrothMark)* Minor discussion of autoaoff.

Added some additional indexing commands.
Revision 14.8*(2002/06/23 20:39:45; WrothMark)* Modified the definitions of the `thebibliography` and `theindex` environments to deal correctly with the bookmarks and table of contents by including entries in both places.
Revision 14.7*(2002/06/23 20:23:40; WrothMark)* Added discussion of confidence intervals.
Revision 14.6*(2002/06/21 02:47:44; WrothMark)*

Added comment lines to the beginnings of functions and programs giving their argument lists.

Added an example of manipulating an arbitrary PMF for a discrete random variable. Converted the bibliography to a specially designed BibTEX format.

Added an experimental Tool Bar menu.

Added general discussion at the beginning of the document.

Added an example on the used of the calculator integration function to find the expected value using the definition.

Revision 14.3*(2002/06/18 21:06:11; WrothMark)* Fixed a typo in the discussion of the check computations for the Equation 7.10 versus 7.11 debate.
Revision 14.2*(2002/06/18 21:02:02; WrothMark)* Changed `binomCI` to use Devore 7.10 rather than 7.11 and midified the discussion for that program accordingly.

Worked some on formatting of `BinomGUI` to reduce the number of lines that run off the page in the printed document. Since there does not appear to be a way to continue a line in the ASCII programming environment without awkward consequences in the execution environment, I have elected to add statements where needed to allow the lines to be shortened.
Revision 14.1*(2002/06/18 19:12:46; WrothMark)* Added various illustrations, including a cartoon from the Pointer (with artist's permission).
Revision 14.0*(16 JUN 02; MBW)* Converted the entire document to `Nuweb` and LaTeX, in order to take advantage of the ASCII export functions of `TI-Graph Link` and to avoid the problems observed with missing characters in the HTML and RTF versions produced from the DocBook implementation.

This also allows use of the various indexing functions in the literate programming tool *Nuweb* and in LaTeX. This revision puts the mechanisms in place for this, but does not take full advantage of them.
Revision 12*(20 June 2001; MBW)* Added `chi2int` discussion.
Revision 11*(20 June 2000; MBW)* Additional general discussion.
Revision 10*(19 June 2000; MBW)* Added an `MA206()` program which sets up a custom menu allowing commonly used functions to be pasted into the entry line.
Revision 9*(17 June 2000; MBW)* Added appendix showing symbols, and procedure for normal probability plots. Included RevHistory in the hypergeo program listing as an experiment; as set up in the print version, the data remains in the SGML file, but is suppressed in the printed version, exactly as hoped.
Revision 8*(15 June 2000; MBW)* Added procedures for graphical descriptive measures. Some update of other areas.
Revision 7*(14 June 2000; MBW)* Added documentation for the `hypergeo()` program, and added the use of the callout element for documentation.
Revision 6*(13 June 2000; MBW)* Completed reference page entries for the four functions documented in this paper, including adding examples. Moved the function definitions to the reference pages, rather than in the general discussion, for better parallelism with the pre-defined functions.

Revision 5*(13 June 2000; MBW)* Added reference entries for hypergeometric distribution. The reference entries for the other defined programs still need to be done.
Revision 4*(12 June 2000; MBW)* Added function definitions for hypergeometric distribution.
Revision 3*(8 June 2000; MBW)* Added definitions for the uniform and exponential CDFs as example user defined functions.
Revision 2*(4 June 2000; MBW)* Added basic information on the functions built in to the statistics application. Very limited coverage of the sections on estimation and hypothesis testing.
Revision 1*(3 June 2000; MBW)* Initial conversion to SGML, including only the chapter on upgrading the calculator.

Appendix E

Upgrading a TI-89 Calculator for MA206 Probability and Statistics

E.1 Overview

To load the TI-89 Statistics with List Editor flash application, you need to load the Advanced Mathematics Software Operating System/base code, and then install the Statistics with List Editor application. The TI manual indicates that it is about four times faster to do this from calculator to calculator than from desktop to calculator.

This discussion assume you have two TI-89 calculators, one with the Advanced Mathematics Software and Statistics with Lists application installed, and one which you are upgrading to that configuration, and that you have a calculator to calculator link cable.

E.2 Installing the Advanced Mathematics Software using another TI-89

1. Ensure both calculators have fresh batteries.

Warning

A power loss (or any other interruption) during this operation will mean the receiving unit has to be reloaded using a computer.

2. Ensure any data which is to be retained on the receiving calculator is backed up to another calculator or computer.

Warning

> This procedure will delete all user variables and reset the receiving calculator to its factory state. This may include deleting flash applications.

3. Link the two TI-89s using the calculator to calculator cable (as described on page 366 of the *TI-89 and TI-92 Plus Guidebook*.

4. On both calculators, select the LINK menu

   (a) Select **[2nd] VAR-LINK**

   (b) Select **F-3 LINK**.

5. On the receiving calculator, select Receive Product Code

   (a) Cursor down until option 5: Receive Product Code is highlighted

   (b) Press **ENTER**

   (c) A warning message will display. Press **ENTER** to continue (or ESC to abort).

6. On the sending calculator, select Send Product Software

   (a) Cursor down until option 4:Send Product SW is highlighted

   (b) Press `Enter`

   (c) A warning message will display. Press **ENTER** to continue (or **ESC** to abort).

7. After a short pause (about five seconds), the receiving calculator will display a status message and progress indicator. Wait until the display clears (about six minutes). When the display clears, the transfer is complete.

Warning

Interrupting the transmission will result in the receiving calculator becoming inoperable until it is reloaded from a computer.

8. Reload any backed-up data to be retained on the receiving calculator

For more information on installing base code updates, see Upgrading Product Software (Base Code), beginning on page 373 of the *TI-89 and TI-92 Plus Guidebook*.

E.3 Installing the Statistics with List Editor Flash Application Using Another TI-89

1. Link the two TI-89s using the calculator to calculator cable (as described on page 366 of the *TI-89 and TI-92 Plus Guidebook* [11].

2. On the sending calculator, select the `LINK` menu by selecting `[2nd ] VAR-LINK`

3. On the sending calculator, select the Stats/List Edi flash application

   (a) Select `F-7 Flashapp` to display the list of flash applications

   (b) Highlight the Stats/List Edi application (it may already be highlighted, for example if its the only one there).

   (c) Press `F-4` to check mark the Stats/List Editor application. A small check mark should appear next to the application name.

4. On the receiving calculator, select the LINK menu by selecting `[2nd] VAR-LINK`. Both calculators should now be in the VAR-LINK screen.

5. On the receiving calculator, select the receive option

   (a) Select **F3 LINK**

   (b) Move the highlight to option 2:Receive

   (c) Press Enter. The messages VAR-LINK WAITING TO RECEIVE and BUSY should appear on the status line.

6. On the sending calculator, select the `Send to TI-89/92 Plus` option

(a) Select **F3 LINK**

(b) Move the highlight to option 3: Send to TI-89/92 Plus

(c) Press **ENTER**.

The message SENDING TISTATLE, a progress bar, and the BUSY indicator should be displayed on the receiving calculator.

7. Wait until the screen clears on the receiving calculator (about 75 seconds). When the receiving calculators VAR-LINK screen returns, the transmission is complete.

For more information on installing flash applications, see Transmitting Variables, Flash Applications, and Folders, beginning on page 367 of the *TI-89 and TI-92 Plus Guidebook*

E.4 Revision History

This is a revision history of the procedure described in this Appendix (i.e. of the procedure for upgrading the TI-89).
Revision 1.2*(3 June 2000; MBW)* Conversion to SGML as a chapter in a DocBook book.
Revision 1.1*(1 June 2000; MBW)* Minor edits after testing the procedunes.
Revision 1.0*(30 May 2000; MBW)* Initial version.

Appendix F

Document Production Notes

This document is produced using *Nuweb*, a literate programming application originally by Preston Briggs and now maintained on SourceForge. The PDF file is produced using pdfLaTeX; customizations for this document are in the file `calculator.sty` shown below.

The intitial part of the style file is identification of the LaTeX package.

"calculator.sty" 89a ≡

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{calculator}[2002/06/23 MA206 Calculator Notes]
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

Now we load the packages that are used in the document.

"calculator.sty" 89b ≡

```
\usepackage{pxfonts}
\usepackage{alltt,graphicx,longtable,makeidx,RCS,verbatim}
\usepackage{hyperref}
\usepackage{acronym}
\usepackage{multicol}
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

This records the revision of the document that produces this.

"calculator.sty" 90a ≡

```
\RCS$Revision: 15.1 $
\RCS$Date: 2002/09/29 20:19:10 $
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

Now we set up the PDF annotations.

"calculator.sty" 90b ≡

```
\hypersetup{pdftitle={TI-89 in MA206},
            pdfkeywords={probability,
                         statistics,
                         calculator,
                         TI-89},
            pdfpagemode={UseOutlines},
            hypertexnames=false }
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

This environment will allow us to write a BibTEX bibliography as part of this file. It will be written to disk as ti89ma206.bib, and then can be processed using the normal BibTEX commands.

"calculator.sty" 90c ≡

```
\newwrite\verbatim@out
\newenvironment{bibtex}%
 {\@bsphack
  \immediate\openout \verbatim@out \jobname.bib
  \let\do\@makeother\dospecials\catcode`\^^M\active
  \def\verbatim@processline{%
    \immediate\write\verbatim@out{\the\verbatim@line}}%
  \verbatim@start}%
 {{\immediate\closeout\verbatim@out\@esphack}}
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

Likewise we modify the definition of the `thebibliography` environment to cause it to make an appropriate table of contents (and hence PDF bookmark) entry.

"`calculator.sty`" 91a ≡

```
\setlength\bibindent{1.5em}
\renewenvironment{thebibliography}[1]
     {\list{\@biblabel{\@arabic\c@enumiv}}%
           {\settowidth\labelwidth{\@biblabel{#1}}%
            \leftmargin\labelwidth \advance\leftmargin\labelsep
            \@openbib@code
            \usecounter{enumiv}%
            \let\p@enumiv\@empty
            \renewcommand\theenumiv{\@arabic\c@enumiv}}%
      \sloppy \clubpenalty4000 \@clubpenalty \clubpenalty
      \widowpenalty4000%
      \sfcode`\.\@m} {\def\@noitemerr
       {\@latex@warning{Empty `thebibliography' environment}}%
      \endlist} \let\@openbib@code\@empty
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

This change redefines the `index` environment so that it too is appropriately entered into the table of contents and list of bookmarks.

"`calculator.sty`" 91b ≡

```
\renewcommand{\indexname}{General Index}
%  from report.cls
\renewenvironment{theindex} {\begin{multicols}{2} \let\item\@idxitem
               } {\end{multicols}}
               \renewcommand\@idxitem{\par\hangindent 40\p@}
               \renewcommand\subitem{\@idxitem \hspace*{20\p@}}
               \renewcommand\subsubitem{\@idxitem \hspace*{30\p@}}
               \renewcommand\indexspace{\par \vskip 10\p@
               \@plus5\p@ \@minus3\p@\relax}
               \renewcommand*\see[2]{\emph{\seename} #1}
               \renewcommand{\hyperpage}[1]{{#1}}
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

The following macros are used for simplicity in writing.

"calculator.sty" 92a ≡

```
\newcommand{\key}[1]{\texttt{#1}}
\newenvironment{Note}{%
                \begin{quotation}
                \begin{center}Note\end{center}\noindent}%
               {\end{quotation}}
\newenvironment{Warning}{\begin{center}{Warning}\end{center}%
                \begin{quotation}\noindent}{%
               \end{quotation}}
\newenvironment{tip}{%
                \begin{quotation}\noindent\textbf{Tip: }}{%
               \end{quotation}}
\newenvironment{varlist}{%
 \begin{description}
 \renewcommand{\key}[1]{\item[\texttt{##1}\index{##1 (variable)}]}}{%
\end{description}}
◇
```

File defined by 89ab, 90abc, 91ab, 92ab.

"calculator.sty" 92b ≡

```
\newcommand{\dfn}[1]{\textit{#1}\index{#1}}
\newcommand{\app}[1]{\textit{#1}\index{#1}}
\newcommand{\fname}[1]{\texttt{#1}\index{#1 (filename)}}
\newcommand{\guimenu}[1]{\texttt{#1}}
\newcommand{\program}[1]{\texttt{#1}}
\newcommand{\acr}[1]{\ac{#1}\index{#1}}
\newenvironment{revision}[3]{%
\par\noindent\textbf{Revision #1}\textit{(#2; #3)}}{%
 \relax}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand{\StatsLE}{\textsl{TI-89 Statistics with List Editor}}
\newcommand{\phat}{\ensuremath{\hat{p}}}
\newcommand{\qhat}{\ensuremath{\hat{q}}}
\newcommand{\TIint}{\ensuremath{\int}}
```

```
\newcommand{\caret}{\^{\space}}
\def\nCr(#1,#2){\left(\begin{array}{c}#1\\#2\end{array}\right)}
\def\P(#1){\ensuremath{{\cal P}\left(#1\right)}}
\def\V(#1){\ensuremath{{\cal V}\left(#1\right)}}
\def\E(#1){\ensuremath{{\cal E}\left(#1\right)}}
\def\CI(#1,#2){\ensuremath{\left(#1, #2\right)}}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\makeindex
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```
◇

File defined by 89ab, 90abc, 91ab, 92ab.

Appendix G

Acronyms

CAS Computer Algebra System

CDF Cumlative Distribution Function

GUI Graphical User Interface

MVUE Minimum Variance Unbiased Estimator

NIST National Institute of Standards and Technology

PDF Probability Density Function

PMF Probability Mass Function

USMA United States Military Academy

Appendix H

Bibliography

[1] Alan Agresti and Brent Coull. Approximate is better than 'exact' for interval estimation of a binomial proportion. *The American Statistician*, pages 119–126, 1998. Cited in [4].

[2] Preston Briggs and Marc W. Mengel. Nuweb version 1.0b1: A simple literate programming tool, 2002. `http://sourceforge.net/projects/nuweb/`; version 1.0b1 was released 24 February 2002. An older version is available as a Windows executable from `http://www.west-point.org/users/usma1978/36200/LitProg/Nuweb/readme.htm%l`.

[3] Gail Burrill, Jacquie Allison, Glenda Breaux, Signe Kastberg, Keith Leatham, and Wendy Sanchez. *Handheld Graphing Technology in Secondary Mathematics: Research Findings and Implications for Classroom Practice*. Texas Instruments, 2002? Prepared through a grant to Michigan State University.

[4] Jay L. Devore, editor. *Probability and Statistics for Engineering and the Sciences*. Duxbury Press, Belmont, CA, 5th edition, 1999.

[5] David Knellinger. MA 206 – Probability and Statistics. Department of Mathematical Sciences Instructional Memorandum 206-1, 27 May 2002.

[6] Tom Mount. TI 89/92 Basic Programming Manual. Downloaded from `http://www.ticalc.org/pub/text/misc` on 19 JUN 02.

[7] Tom Mount. TI 89/92 Advanced Programming Manual, 2001. Downloaded from `http://www.ticalc.org/pub/text/misc` on 19 JUN 02.

[8] Barry N. Taylor and Chris E. Kuyatt. *Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results*. National Institute of Standards and Technology, Gaithersburg, MD, September 1994.

[9] Texas Instruments. *Advanced Mathematics Software*. This is an overview page for the AMS upgrade.

[10] Texas Instruments. *Statistics with List Editor*. `http://www.ti.com/calc/flash/pdf/statsle.pdf`. This is the users guide for the Statistics with List Editor application.

[11] Texas Instruments. *TI-89 and TI-92 Plus Guidebook for Advanced Mathematics Software Version 2.0*. This is the updated manual for the Advanced Mathematics Software; it replaces the manual included with the calculator.

[12] Texas Instruments Inc. *TI-GRAPH LINK for Windows for the TI-89: Condensed Guidebook*.

[13] Unknown. ASCII conversion codes for ASCII calculator program code. Downloaded from `http://www.ticalc.org/pub/text/misc` on 19 JUN 02. The only indication of authorship is "Last Update: 12 Mar 96 SLR" at the end of the file.

Appendix I

Indexes

I.1 Files Written

I.2 Scraps Defined

I.3 Identifiers

I.4 General Index

To Do

- Convert all of the input and output variable lists to `varlists`
- Add GUI's callable by the MA206 menu for the remaining TI-89 functions commonly used.
- Index the various distributions and add cross references between the discussion of usage and the relevant implementing function.
- Add a section discussing the used of simulation on the calculator, with particular reference to the simulation of sampling experiments and computation of test statistics or confidence intervals.
- Extend the CI functions to allow computations on a sample provided as a list (similar to the TIStat functions).