StInt—A Computer Program for Computing Statistical Intervals

William Q. Meeker
I. Shang Jackson Chow
Department of Statistics
Iowa State University
Ames, IA 50011

December 28, 1993

Abstract

StInt is a computer program that can be used to compute a number of different kinds of statistical intervals.

Key Words: Confidence Interval, Prediction Interval, Tolerance Interval, Algorithm

1 Introduction

1.1 Importance of Statistical Intervals

Many decisions need to make decisions on the basis of limited sample data. Statistical intervals can usefully quantify the uncertainty in our data. Hahn and Meeker (1991) outline assumptions, methods, and applications for different kinds of statistical intervals.

1.2 Types of Statistical Intervals

Various types of statistical intervals can be constructed from sample data. Frequently used intervals are:

- A *confidence interval* to contain a parameter of the sampled population or process. For instance, an investigator might want to capture the mean or the standard deviation of a normal distribution or a function of such parameters or a percentages point (percentile) or the distribution of the probability of a specified event.

- A *tolerance interval* to contain, with a specified degree of confidence, at least a certain proportion of the units in the population.

- A *prediction interval* to predict one or more further observations, or some function of observations.

See Hahn and Meeker (1991) for further discussion of these different kinds of intervals.

1.3 Initial Guidance and Overview

StInt can be used to compute statistical intervals for the normal distribution, the binomial distribution, the Poisson distribution and it can be used to construct distribution-free statistical intervals.

The remainder of this report is organized as follows:

- Section 2 provides two simple introductory examples.

- Section 3 describes modes of operation (interactive and batch).

- Section 4 describes, in detail, all of the **StInt** commands.

- Appendix A provides installation instructions for MS/DOS computers.

- Appendix B contains an index and brief description of the **StInt** commands.

- Appendix C gives **StInt** example input, section-by-section, for most of the examples in Hahn and Meeker (1991). Hahn and Meeker (1991) provide detailed explanations of these examples.

- Appendix E explains some warning and error messages.

2 Some Simple Examples

As an introduction for the new user, and to show easy it is to perform standard analyses, this section contains two simple examples that illustrate some a **StInt**'s basic commands and capabilities. The commands used here, and all of the other **StInt** commands, are described in detail in Section 3.

Example 1: Normal distribution confidence interval and confidence bound

Lifetimes (in hours) of a certain kind of battery are assumed to be normally distributed. In a random sample of size $n = 25$, the sample mean was found to be $\overline{x} = 75$ and the sample standard deviation was $s = 10$. The desired intervals were computed with the following **StInt** commands:

```
ndata s=10.0 xbar=75.0 n=25
ncim
nciy p=.1 kside=1 for lower one-sided interval with conlev=.90
stop
```

The **ndata** command tells **StInt** to read the summary of a set of data which we will assume comes from a normal distribution. The **ncim** command requests a 95% (default) two-sided (default) confidence interval for normal mean. The **nciy** command specifies a one-sided lower (`kside=1`) 90% (`conlev=.90`) confidence bound for normal distribution 10th percentile (.1 quantile).

Example 2: Confidence intervals for a population proportion

An unknown proportion of 1000 installed engines were installed in a nonconforming state and will have to undergo expensive rework. A survey was conducted to estimate the proportion of nonconforming engines. Management wants a confidence interval for π, the proportion of engines that will require rework. From the available data, we have $x = 1$, the observed nonconforming engine from a random sample of size $n = 10$. We want to calculate a 95% confidence interval for π. This analysis was done with the following **StInt** commands:

```
bdata x=1 failure in n=10 trials
bcip
```

```
stop
```

The **bdata** command tells **StInt** to read the summary binomial data. The **bcip** command requests the confidence intervals for a population proportion. By default, the **bcip** command provides a two-sided 95% confidence interval for the binomial proportion.

3 StInt Modes of Operation

StInt can be used in either an interactive mode or in a batch mode. To just do a few simple operations with **StInt**, the interactive model is much simpler. The batch model, however, is useful for doing larger jobs.

3.1 StInt Interactive Mode of Operation

To run **StInt** in the interactive mode, "cd" to the **StInt** directory and type the command,

```
c:\STINT> stint

 w. q. meeker
 department of statistics
 iowa state university
 ames, iowa     50011

   with assistance from jackson chow


 version of december 4, 1993


 ibmpc version

stint >
```

You can then give commands to the **StInt** prompt (`stint >`). To get out of **StInt** , type

```
stint> stop
```

3.2 StInt Batch Mode of Operation

To run **StInt** in the batch mode, you must first specify the input and output files. For example, in MS/DOS, type the commands

```
C:\STINT> set 5=c:\stint\book.exa
C:\STINT> set 6=c:\stint\book.out
```

to define file `BOOK.EXA` as the input file containing **StInt** commands and `BOOK.OUT` as the output file into which results are to be written. Then to run **StInt**

```
C:\STINT> stint
```

When the prompt

```
C:\STINT>
```

returns, the output can be viewed, for example with

```
C:\STINT> more < BOOK.OUT
```

(early versions of MS/DOS may not have the `more` command).

To return the definitions of units 5 and 6 to their original meanings (e.g. to return to interactive use of **StInt**)

```
C:\STINT> set 5=terminal
C:\STINT> set 6=terminal
```

4 StInt Commands

4.1 StInt Command Syntax

Input to **StInt** is accomplished through a sequence of simple commands. This section contains a detailed description of each of the **StInt** commands. Each command begins with a *keyword* (containing from 1 to 4 letters). The keyword must appear in column 1-4. All **StInt** keywords should be given in *lower* case. Numerical arguments (if any) can appear anywhere in the command, separated by spaces or commas; the order of the arguments *is* usually important. Nonnumeric characters can (and should) be used to document the commands, as illustrated in the examples. For example

```
nciy p=.1 kside=1 for lower one-sided interval with conlev=.90
```

is more descriptive than

```
nciy   .1   1   .90
```

In the description of the commands to follow, **I** represents an argument which must be a nonnegative integer, **R** represents a real number.

4.2 General Description of the StInt Commands

The **StInt** commands are divided, by function, into 3 groups.

- Commands To Input Data or Data Summaries.
- Commands for Calculating Statistical Intervals for a Normal Distribution (Chapters 3 and 4 of Hahn and Meeker (1991))
- Commands To Compute Distribution-free Statistical Intervals (Chapter 5 of Hahn and Meeker (1991))
- Commands to Compute Distribution-free Probabilities Associated with Order Statistics (Chapter 5 of Hahn and Meeker (1991))
- Commands for Calculating Statistical Intervals for Proportions and Percentages (Chapter 6 of Hahn and Meeker (1991))
- Commands for Calculating Statistical Intervals for the Number of Occurrences (Chapter 7 of Hahn and Meeker (1991))
- Commands To Compute Distribution Probabilities and Quantiles
- Commands to control program options (confidence level, one or two-sided intervals, etc.)

4.3 Commands To Input Data or Data Summaries

1. **ndata** *xbar*=**R** *s*=**R** *n*=**I**

   This command instructs **StInt** to read a summary of normal data consisting of the sample mean ($\overline{x}$), the sample standard deviation ($s > 0$) and a sample size ($n > 0$). These data can be used subsequently to construct statistical intervals.

   Another method of setting data for the normal distribution is to give a **cndata** command after reading a vector of data with the **read** command.

2. **cndata**

   After reading a vector of observations with the **read** or the **fread** command, **cndata** instructs **StInt** to compute mean ($\overline{x}$) and standard deviation (s). These summary statistics will be used to construct statistical intervals for the normal distribution. No arguments are required. [Also see the **ndata** command.]

3. **fread** n=**I** `observations` from a specified file

   This command instructs **StInt** to read a vector of n (> 0) observations from a specified file. In the interactive mode, the user will be prompted for a file name. In the batch mode, the file name should follow in the record following the **fread** command. After reading this command, **StInt** will read successive 80-column records from the specified file until n numbers have been found. Then **StInt** will return to reading other commands. Multiple observations on a record should be separated by one or more spaces.

4. **read** n=**I** `observations`

   This command instructs **StInt** to read a vector of n (> 0) observations from records (or from the keyboard in the interactive mode) immediately following the command. After reading this command, **StInt** will read successive 80-column records until n numbers have been found. Then it will return to reading other commands. Data in these records should be separated by one or more spaces.

5. **nvec** n=**I** `observations`

   Allows the specification of a dummy sample size to allow one to use the distribution-free interval commands to get order statistic indices, without having to input the entire sample.

6. **bdata** x=**I** `successes in` n=**I** `trials`

   This command instructs **StInt** to read a binomial data summary consisting of the number of successes ($x \geq 0$) and the number of trials ($n > 0$). The command must contain exactly 2 numbers.

7. **pdata summary: x=I occurrences in n=R units of time (or space)**

This command instructs **StInt** to read a Poisson data summary consisting of the number of occurrences ($x \geq 0$) in a known number of units of observed time or space ($xn > 0$).

4.4 Commands for Calculating Statistical Intervals for a Normal Distribution (Chapters 3 and 4 of Hahn and Meeker (1991))

8. **ncim** `[kside=`**I** `[conlev=` **R**`]]`

   This command instructs **StInt** to compute a **n**ormal distribution **c**onfidence **i**nterval for **m**ean of a normal distribution. See Section 4.2 of Hahn and Meeker (1991) for a description of this interval.

   No arguments are required for this command, however normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

9. **ncis** `[kside=`**I** `[conlev=` **R**`]]`

   This command instructs **StInt** to compute a **n**ormal distribution **c**onfidence **i**nterval for **s**tandard deviation. See Section 4.3 of Hahn and Meeker (1991) for a description of this interval.

   No arguments are required, however normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

10. **nciy** *p*=**R** `[kside=`**I** `[conlev=` **R**`]]`

    This command instructs **StInt** to compute a **n**ormal distribution **c**onfidence **i**nterval for the $100p$ percentile. See Section 4.4 of Hahn and Meeker (1991) for a description of this interval.

    Normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

11. **npgt** *yprime*=**R** `[kside=`**I** `[conlev=` **R**`]]`

    This command instructs **StInt** to compute a **n**ormal distribution **c**onfidence **i**nterval for the **n**ormal distribution **p**robability of being **g**reater **t**han y'. See Section 4.5 of Hahn and Meeker (1991) for a description of this interval.

Normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

12. **nplt** *yprime*=**R** [kside=**I** [conlev= **R**]]

    This command instructs **StInt** to compute a **n**ormal distribution **c**onfidence **i**nterval for the **n**ormal distribution **p**robability of being **l**ess **t**han y'. See Section 4.5 of Hahn and Meeker (1991) for a description of this interval.

    Normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

13. **nti** p=**R** [kside=**I** [conlev= **R**]]

    This command instructs **StInt** to compute a **n**ormal distribution **t**olerance **i**nterval to enclose $100p\%$ of the distribution. See Section 4.6 of Hahn and Meeker (1991) for a description of this interval.

    Normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

14. **npim** m=**I** [kside=**I** [conlev= **R**]]

    This command instructs **StInt** to compute a **n**ormal distribution **p**rediction **i**nterval for the sample **m**ean $\overline{y}$ of m future sample observations from the same population or process. See Section 4.7 of Hahn and Meeker (1991) for a description of this interval.

    Normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

15. **npik** m=**I** [kside=**I** [conlev= **R**]]

    This command instructs **StInt** to compute an approximate **n**ormal distribution simultaneous **p**rediction **i**nterval to contain *all* of $k = m$ future sample observations from the same population or process (based on a Bonferroni approximation). See Section 4.8 of Hahn and Meeker (1991) for a description of this interval.

Normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

16. **npis** m=**I** [kside=**I** [conlev= **R**]]

    This command instructs **StInt** to compute a **n**ormal distribution **p**rediction **i**nterval for the sample **S**tandard deviation s of m future sample observations from the same population or process. See Section 4.9 of Hahn and Meeker (1991) for a description of this interval.

    Normal data must be in place (use **ndata** or **cndata**). If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

17. **oot1** n=**I**, p= **R**, $gamma$= **R**

    This command instructs **StInt** to compute a factor from Table 1 of Odeh and Owen (1980). See Chapter 4 of Hahn and Meeker (1991) for applications.

18. **oot7** n=**I**, xk= **R**, eta= **R**

    This command instructs **StInt** to compute a factor from Table 7 of Odeh and Owen (1980). See Chapter 4 of Hahn and Meeker (1991) for applications.

4.5 Commands To Compute Distribution-free Statistical Intervals (Chapter 5 of Hahn and Meeker (1991))

The endpoints of distribution-free statistical intervals are chosen to be particular ordered observations. For example, a two-sided conservative $100(1-\alpha)\%$ distribution-free statistical interval is obtained from a given sample of size n as $[Y_{(\ell)}, Y_{(u)}]$, where ℓ and u are are integer indices indicating particular ordered observations and $Y_{(i)}$ is the ith ordered observation. The indices ℓ and u depend on the type of interval and on the specified confidence level $100(1-\alpha)\%$.

A one-sided $100(1-\alpha)\%$ distribution-free lower statistical bound is obtained as $Y_{(\ell)}$. Similarly, a one-sided distribution-free $100(1-\alpha)\%$ upper statistical bound is obtained as $Y_{(u)}$. The commands in this section are used to obtain the order statistic indices for distribution-free statistical intervals. The indices ℓ and u for these observations depend on the particular interval and the confidence level. Chapter 5 of Hahn and Meeker (1991) provides more information about these intervals.

19. **ciyp** `for quantile` p=**R** `[kside=`**I** `[conlev=` **R**`]]`

    This command asks **StInt** to compute a distribution-free **c**onfidence **i**nterval for a distribution $100p$ **p**ercentile Y_P. See Section 5.2 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. If data are in place (from the **read** or **fread** commands, **ciyp** will return the indices of the particular order statistics and the order statistics that provide the interval.

    A two-sided conservative $100(1-\alpha)\%$ distribution-free confidence interval for Y_p, the $100p$th percentile of the sampled population, is obtained from a given sample of size n as $[Y_{(\ell)}, Y_{(u)}]$, where ℓ and u are are integer indices indicating particular ordered observations and $Y_{(i)}$ is the ith ordered observation. The indices ℓ and u are functions of the specified confidence level $100(1-\alpha)\%$ and p.

    A one-sided $100(1-\alpha)\%$ distribution-free lower confidence bound for the $100p$th percentile of the population is obtained as $Y_{(\ell)}$. Similarly, a one-sided distribution-free $100(1-\alpha)\%$ upper confidence bound for the $100p$th percentile of the population is obtained as $Y_{(u)}$.

20. **dfti** `to contain a proportion` p=**R** `[kside=`**I** `[conlev=` **R**`]]`

    This command instructs **StInt** to compute a **d**istribution-**f**ree **t**olerance **i**nterval to contain a specified percentage of a population. See Section 5.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. If data are in place (from the **read** or **fread** commands), **npti** will return the indices of the particular order statistics and the order statistics that provide the interval.

    Note that a distribution-free one-sided tolerance bound is equivalent to a one-sided distribution-free confidence bound for a percentile of that population. That is, a one-sided distribution-free lower (upper) $100(1-\alpha)\%$ tolerance bound that will be exceeded by (that will exceed) at least $100p\%$ of the population, is the same as a distribution-free lower (upper) $100(1-\alpha)\%$ confidence bound for the $100p$th percentile of the population (see the **ciyp** command).

21. **pikm** `enclose` k=**I** `of` m=**I** `future observations [kside=`**I** `[conlev=` **R**`]]`

This command instructs **StInt** to compute a distribution-free **p**rediction **i**nterval to contain k of m ordered observations from a future sample. See Section 5.4 of Hahn and Meeker (1991) for a description of this interval.

If data are in place (from the **read** or **fread** commands), **pikm** will return the indices of the order statistics and the particular order statistics that provide the interval. Otherwise, if the sample size has been specified with the **nvec** command, only the order statistics indices are provided. If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

22. **piyj** `enclose the` j=**I** `ordered observation of` m=**I** `future observations [kside=`**I** `[conlev=` **R**`]]`

This command instructs **StInt** to compute a distribution-free **p**rediction **i**nterval to contain $y_{(j)}$, the jth ordered observation from a future sample of m observations. See Section 5.5 of Hahn and Meeker (1991) for a description of this interval.

If data are in place (from the **read** or **fread** commands), **piyj** will return the indices of the order statistics and the particular order statistics that provide the interval. Otherwise, if the sample size has been specified with the **nvec** command, only the order statistics indices are provided. If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters.

4.6 Commands to Compute Distribution-free Probabilities Associated with Order Statistics (Chapter 5 of Hahn and Meeker (1991))

23. **epyp** n=**I**,ℓ=**I**,u=**I**,p=**R**

Evaluate the distribution-free probability that the 100pth percentile will be enclosed by the order statistics $x_{(\ell)}$ and $x_{(u)}$ of n observations. See Section 5.2 of Hahn and Meeker (1991).

24. **epti** n=**I**,ℓ=**I**,u=**I**,p=**R**

Evaluate the distribution-free probability that 100p% of the population will be enclosed by the order statistics $x_{(\ell)}$ and $x_{(u)}$ of n observations. See Section 5.3 of Hahn and Meeker (1991).

25. **epkm** n=**I**,ℓ=**I**,u=**I**,m=**I**,k=**I**

    Evaluate the distribution-free probability of having at least k of m future observations fall between the order statistics $x_{(\ell)}$ and $x_{(u)}$ of n past observations. See Section 5.4 of Hahn and Meeker (1991).

26. **epyj** n=**I**,ℓ=**I**,u=**I**,m=**I**,j=**I**

    Evaluate the distribution-free probability of having the jth order statistics $y_{(j)}$ of m future observations lie between the order statistics $x_{(\ell)}$ and $x_{(u)}$ of n past observations. See Section 5.5 of Hahn and Meeker (1991).

4.7 Commands for Calculating Statistical Intervals for Proportions and Percentages (Chapter 6 of Hahn and Meeker (1991))

27. **bcip** [kside=**I** [conlev= **R**]]

    This command asks **StInt** to compute a **b**inomial **c**onfidece **i**nterval for the **p**roportion π. No arguments are required. See Section 6.2 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that binomial data be in place (see **bdata**).

28. **bplt** less than J= **I** in m= **I** future trials [kside=**I** [conlev= **R**]]

    Compute a confidence interval (or bound) on the **b**inomial **p**robability of being **l**ess **t**han J in a future sample of m units. See Section 6.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that binomial data be in place (see **bdata**).

29. **bple** less than or equal to J= **I** in m= **I** future trials [kside=**I** [conlev= **R**]]

Compute a confidence interval (or bound) on the **b**inomial **p**robability of being **l**ess than or **e**qual to J in a future sample of m units. See Section 6.3 of Hahn and Meeker (1991) for a description of this interval.

If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that binomial data be in place (see **bdata**).

30. **bpgt** `greater than` J`=` **I** `in m=` **I** `future trials [kside=`**I** `[conlev=` **R**`]]`

    Compute a confidence interval (or bound) on the **b**inomial **p**robability of being **g**reater **t**han J in a future sample of m units. See Section 6.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that binomial data be in place (see **bdata**).

31. **bpge** `greater than or equal to` J`=` **I** `in m=` **I** `future trials [kside=`**I** `[conlev=` **R**`]]`

    Compute a confidence interval (or bound) on the **b**inomial **p**robability of being **g**reater than or **e**qual to J in a future sample of m units. See Section 6.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that binomial data be in place (see **bdata**).

32. **btb** p=**R** m `=` **I** `[kside=`**I** `[conlev=` **R**`]]`

    This command asks **StInt** to compute a **b**inomial **t**olerance **b**ound to exceed (be exceeded by) y, the number of nonconforming units in $100p$ percent of future batches of size m units. See Section 6.4 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that binomial data be in place (see **bdata**).

33. **bpi** `for` m`=` **I** `future trials [kside=`**I** `[conlev=` **R**`]]`

    This command tells **StInt** to compute a **b**inomial **p**rediction **i**nterval on the number of nonconforming units y in a future sample of size m.

See Section 6.5 of Hahn and Meeker (1991) for a description of this interval.

If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that binomial data be in place (see **bdata**).

4.8 Commands for Calculating Statistical Intervals for the Number of Occurrences (Chapter 7 of Hahn and Meeker (1991))

34. **pcir** `[kside=`**I** `[conlev=` **R**`]]`

    This command tells **StInt** to compute a **P**oisson **c**onfidence **i**nterval for the Poisson occurrence **r**ate. See Section 7.2 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that Poisson data be in place (see **pdata**).

35. **pplt** `less than` J`=` **I** `in` m`=` **I** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`

    Compute a confidence interval (or bound) on the **P**oisson **p**robability of being **l**ess **t**han J in m= **I** future units of time or space. See Section 7.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that Poisson data be in place (see **pdata**).

36. **pple** `less than or equal to` J`=` **I** `in` m`=` **I** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`

    Compute a confidence interval (or bound) on the **P**oisson **p**robability of being **l**ess than or **e**qual to J in m= **I** future units of time or space. See Section 7.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that Poisson data be in place (see **pdata**).

37. **ppgt** `greater than` J= **I** `in` m= **I** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`

    Compute a confidence interval (or bound) on the **P**oisson **p**robability of being **g**reater **t**han J in m= **I** future units of time or space. See Section 7.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that Poisson data be in place (see **pdata**).

38. **ppge** `greater than or equal to` J= **I** `in` m= **I** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`

    Compute a confidence interval (or bound) on the **P**oisson **p**robability of being **g**reater **t**han or equal to J in m= **I** future units of time or space. See Section 7.3 of Hahn and Meeker (1991) for a description of this interval.

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that Poisson data be in place (see **pdata**).

39. **ptb** p=**R** m=**R** `[kside=`**I** `[conlev=` **R**`]]`

    This command instructs **StInt** to compute a one-sided **P**oisson **t**olerance **b**ound to exceed (be exceeded by) the number of occurrences in a specified period of m units of time or space in at least $100p\%$ of the future trials. See Section 7.4 of Hahn and Meeker (1991).

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that Poisson data be in place (see **pdata**).

40. **ppi** `for` m=**R** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`

    This command asks **StInt** to compute a **P**oisson **p**rediction **i**nterval for the number of occurrences in a future observation in m (> 0) units of time or space, based on the results of previous sample. See Section 7.5 of Hahn and Meeker (1991).

    If specified, the optional **kside** and **conlev** arguments override the current defaults for these parameters. This command requires that Poisson data be in place (see **pdata**).

4.9 Commands To Compute Distribution Probabilities and Quantiles

The following commands can be used to compute entries from the commonly used distributions. These commands provide more flexibility than Tables 5, 6, 7, 8, and 9 in Hahn and Meeker (1991), eliminating any need for interpolation.

41. **tpro** *dof*= **I**,*x*= **R**

    Compute the t-distribution probability $\Pr(T < x)$ where T follows a t-distribution with *dof* degrees of freedom.

42. **tqua** *p*= **R** *dof*= **I**

    Compute the t-distribution p quantile ($100p$ percentile). That is, give t_p such that $\Pr(T < t_p) = p$ where T follows a t-distribution with *dof* degrees of freedom.

43. **fpro** *ndof*= **I**, *ddof*= **I**, *x*= **R**

    Compute the F-distribution probability $\Pr(F < x)$ where where F follows a F-distribution with *ndof* numerator degrees of freedom and *ddof* denominator degrees of freedom.

44. **fqua** *p*= **R** *ndof*= **I**, *ddof*= **I**

    Compute the F-distribution p quantile ($100p$ percentile). That is, give f_p such that $\Pr(F < f_p) = p$ where F follows a F-distribution with *ndof* numerator degrees of freedom and *ddof* denominator degrees of freedom.

45. **chpr** *dof*= **I**,*x*= **R**

    Compute the χ^2-distribution probability $\Pr(X < x)$ where X follows a χ^2-distribution with *dof* degrees of freedom.

46. **chqu** *p*= **R** *dof*= **I**

    Compute the χ^2-distribution p quantile ($100p$ percentile). That is, give x_p such that $\Pr(X < x_p) = p$ where X follows a χ^2-distribution with *dof* degrees of freedom.

47. **npro** *z*= **R**

    Compute the normal distribution probability $\Pr(Z < z)$ where Z follows a standard normal distribution.

48. **nqua** p= **R**

    Compute the normal distribution p quantile ($100p$ percentile). That is, give z_p such that $\Pr(Z < z_p) = p$ where Z follows a standard normal distribution.

49. **bino** x=**I**,n=**I**,p= **R**

    Compute the binomial probability of x successes in n trials with success probability on np.

50. **pois** x=**I**, n=**I**, *lambda*= **R**

    Compute the Poisson probability of x occurrences in n units of time or space with mean rate λ per unit of time or space.

4.10 Commands To Control Program Options

51. **batch**

    This command tells **StInt** to go into the batch mode (turns off the interactive prompt)

52. **interactive**

    This command tells **StInt** to go into the interactive mode (turns on the interactive prompt). Stint always starts in the interactive mode.

    This command allows one to specify an alternate default **con**fidence **lev**el of **R** percent provided by **StInt**. The initial default is 95 percent.

53. `conlev=`**R** `percent`

    This command allows one to specify an alternate default **con**fidence **lev**el of **R** percent provided by **StInt**. The initial default is 95 percent.

54. **kside=I**

    This command instructs **StInt** to choose an alternate default for kside which determines if a bound or interval should be lower, upper, or two-sided. One of the following intervals should be indicated by choosing the appropriate value of **I**:
    1 for a one-sided lower bound
    2 for a one-sided upper bound
    3 for a two-sided interval

55. **c**· · ·

    This command can be used to insert comments among other **StInt** commands. The **c** must be followed by 3 spaces before the beginning of the comment.

56. **stop**

    This command indicates the end of a **StInt** program run.

57. **dump** `level=`**I**

    This command instructs the program to print out internal program variables indicating status. This command is used primarily for debugging.

58. **help**

    This command asks **StInt** to print out a list of all stint commands.

A Installation Instructions for MS/DOS Computers

To install **StInt**, it is suggested that you create a separate directory for the **StInt** files and copy the files from the supplied diskette to this directory. The files include the **StInt** executable, example commands (set up to run in the batch mode) and some example data. **StInt** can be run from this directory or the directory could be added to the path by making the appropriate change on the autoexec.bat file. To install **StInt**, put the **StInt** diskette in the disk drive (say drive A) and do the following

```
C:> mkdir stint
C:> cd stint
C:\STINT> copy a:*.* *.*
```

Then to run **StInt**, refer to the instructions in Section 3. For example,

```
C:\STINT> stint
```

B Index Of StInt Commands

This appendix contains a list of the **StInt** commands, organized into groups according to function. Section 4 provides details on the use of these commands.

B.1 Commands To Input Data or Data Summaries

1. **ndata** *xbar*=**R** *s*=**R** *n*=**I**
2. **cndata**
3. **fread** *n*=**I** `observations` from a file
4. **read** *n*=**I** `observations`
5. **nvec** *n*=**I** `observations`
6. **bdata** *x*=**I** `successes in` *n*=**I** `trials`
7. **pdata** `summary:` *x*=**I** `occurrences in` *n*=**R** `units of time (or space)`

B.2 Commands for Calculating Statistical Intervals for a Normal Distribution (Chapters 3 and 4 of Hahn and Meeker (1991))

8. **ncim** `[kside=`**I** `[conlev=` **R**`]]`
   Normal distribution confidence interval for the mean.
9. **ncis** `[kside=`**I** `[conlev=` **R**`]]`
   Normal distribution confidence interval for the standard deviation.
10. **nciy** *p*=**R** `[kside=`**I** `[conlev=` **R**`]]`
    Normal distribution confidence interval for a percentile.
11. **npgt** *yprime*=**R** `[kside=`**I** `[conlev=` **R**`]]`
    Normal distribution confidence interval for the probability greater than.
12. **nplt** *yprime*=**R** `[kside=`**I** `[conlev=` **R**`]]`
    Normal distribution confidence interval for the probability less than.
13. **nti** *p*=**R** `[kside=`**I** `[conlev=` **R**`]]`
    Normal distribution tolerance interval.
14. **npim** *m*=**I** `[kside=`**I** `[conlev=` **R**`]]`
    Normal distribution prediction interval for a sample mean.
15. **npik** *m*=**I** `[kside=`**I** `[conlev=` **R**`]]`
    Normal distribution simultaneous prediction interval for all k of m future observations.

16. **npis** m=**I** `[kside=`**I** `[conlev=` **R**`]]`
    Normal distribution prediction interval for a future sample standard deviation.

17. **oot1** n=**I**, p= **R**, $gamma$= **R**
    Odeh and Owen Table One.

18. **oot7** n=**I**, xk= **R**, eta= **R**
    Odeh and Owen Table Seven.

B.3 Commands To Compute Distribution-free Statistical Intervals (Chapter 5 of Hahn and Meeker (1991))

19. **ciyp** `for quantile` p=**R** `[kside=`**I** `[conlev=` **R**`]]`
    Distribution-free confidence for the $100p$ percentile (p quantile) of a distribution.

20. **dfti** `to contain a proportion` p=**R** `[kside=`**I** `[conlev=` **R**`]]`
    Distribution-free tolerance interval for a distribution.

21. **pikm** `enclose` k=**I** `of` m=**I** `future observations [kside=`**I** `[conlev=` **R**`]]`
    Distribution-free prediction interval for k of m future observations.

22. **piyj** `enclose the` j=**I** `ordered observation of` m=**I** `future observations [kside=`**I** `[conlev=` **R**`]]`
    Distribution-free prediction interval for the jth ordered observation from m future observations.

B.4 Commands to Compute Distribution-free Probabilities Associated with Order Statistics (Chapter 5 of Hahn and Meeker (1991))

23. **epyp** n=**I**,ℓ=**I**,u=**I**,p=**R**

24. **epti** n=**I**,ℓ=**I**,u=**I**,p=**R**

25. **epkm** n=**I**,ℓ=**I**,u=**I**,m=**I**,k=**I**

26. **epyj** n=**I**,ℓ=**I**,u=**I**,m=**I**,j=**I**

B.5 Commands for Calculating Statistical Intervals for Proportions and Percentages (Chapter 6 of Hahn and Meeker (1991))

27. **bcip** `[kside=`**I** `[conlev=` **R**`]]`
    Confidence interval for a binomial distribution proportion.

28. **bplt** `less than` J= **I** `in` m= **I** `future trials [kside=`**I** `[conlev=` **R**`]]`
    Confidence for binomial probability.

29. **bple** `less than or equal to` J= **I** `in` m= **I** `future trials [kside=`**I** `[conlev=` **R**`]]`
    Confidence for binomial probability.

30. **bpgt** `greater than` J= **I** `in m=` **I** `future trials [kside=`**I** `[conlev=` **R**`]]`
    Confidence for binomial probability.

31. **bpge** `greater than or equal to` J= **I** `in m=` **I** `future trials [kside=`**I** `[conlev=` **R**`]]`
    Confidence for binomial probability.

32. **btb** p=**R** m `=` **I** `[kside=`**I** `[conlev=` **R**`]]`
    Binomial tolerance bound.

33. **bpi** `for` m= **I** `future trials [kside=`**I** `[conlev=` **R**`]]`
    Binomial prediction interval.

B.6 Commands for Calculating Statistical Intervals for the Number of Occurrences (Chapter 7 of Hahn and Meeker (1991))

34. **pcir** `[kside=`**I** `[conlev=` **R**`]]`
    Confidence interval for a Poisson distribution occurrence rate.

35. **pplt** `less than` J= **I** `in` m= **I** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`
    Confidence interval for Poisson probability.

36. **pple** `less than or equal to` J= **I** `in` m= **I** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`
    Confidence interval for Poisson probability.

37. **ppgt** `greater than` J= **I** `in m=` **I** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`
    Confidence interval for Poisson probability.

38. **ppge** `greater than or equal to` J= **I** `in` m= **I** `future trials [kside=`**I** `[conlev=` **R**`]]`
    Confidence interval for Poisson probability.

39. **ptb** p=**R** m=**R** `[kside=`**I** `[conlev=` **R**`]]`
    Poisson distribution tolerance bound.

40. **ppi** `for` m=**R** `future units of time or space [kside=`**I** `[conlev=` **R**`]]`
    Poisson distribution prediction interval.

B.7 Commands To Compute Distribution Probabilities and Quantiles

41. **tpro** dof= **I**,x= **R**
    t-distribution probability.

42. **tqua** p= **R** dof= **I**
    t-distribution quantile.

43. **fpro** $ndof$= **I**, $ddof$= **I**, x= **R**
    F-distribution probability.

44. **fqua** p= **R** $ndof$= **I**, $ddof$= **I**
    F-distribution quantile.

45. **chpr** dof= **I**,x= **R**
    Chisquare distribution probability.

46. **chqu** p= **R** dof= **I**
    Chisquare distribution quantile.

47. **npro** z= **R**
    Normal distribution probability.

48. **nqua** p= **R**
    Normal distribution quantile.

49. **bino** x=**I**,n=**I**,p= **R**
Binomial distribution probability.

50. **pois** x=**I**, n=**I**, $lambda$= **R**
Poisson distribution probability.

B.8 Commands To Control Program Options

51. **batch**
Go into the batch mode.

52. **interactive**
Go into the interactive mode.

53. **conlev=R** `percent`
Change default confidence level.

54. **kside=I**
Change default interval/bound.

55. **c**$\cdots$
Comment.

56. **stop**

57. **dump** `level=`**I**

58. **help**

This command asks **StInt** to print out a list of all stint commands.

C StInt Example Input for Examples in Hahn and Meeker (1991)

This Appendix provides a list of input commands that were used to do most of the examples in Hahn and Meeker (1991). In all of the examples, the `kside` (choosing lower bound, upper bound, or two-sides intervals) and `conlev` arguments were specified explicitly, to avoid confusion in checking book results. Detailed explanations of the examples are contained in Hahn and Meeker (1991), in the indicated sections.

```
c  #
c  #chapter 3
c  #
ndata xbar=50.10 sd=1.31 n=5
c  #section 3.3.2
ncim kside=3 conlev=0.95
nti p=0.99 kside=3 conlev=0.95
npik k=10 m=10 kside=3 conlev=0.95
npim m=5 kside=3 conlev=.95
ncis kside=3 conlev=0.95
npis m=5 kside=3 conlev=0.95
c  #
c  #section 3.4.3
c  #
ncim kside=1 conlev=0.95
nti p=0.99 kside=2 conlev=0.95
npik k=10 m=10 kside=1 conlev=0.95
npim m=5 kside=1 conlev=.95
ncis kside=2 conlev=0.95
npis m=5 kside=2 conlev=0.95
c  #
c  #chapter 4
c  #
c  #section 4.2.2
ncim kside=3 conlev=0.95
ncim kside=2 conlev=0.95
c  #section 4.3.2
ncis kside=3 conlev=0.95
ncis kside=2 conlev=0.95
c  #section 4.4.2
nciyp p=.10 kside=3, conlev=0.95
nciyp p=.10 kside=1, conlev=0.95
c  #section 4.5.2
npgt k=48 kside=3, conlev=0.95
npgt k=48 kside=1, conlev=0.95
nplt k=48 kside=2, conlev=0.95
c  #section 4.6.4
nti  p=0.90 kside=3, conlev=0.95
nti p=0.90  kside=2, conlev=0.95
```

```
c  #section 4.7.2
npim m=3 kside=3 and conlev=0.95
npim m=3 kside=2 and conlev=0.95
c  #section 4.8.3
npik k=10, m=10, kside=3, conlev=0.95
npik k=10, m=10, kside=2, conlev=0.95
c  #section 4.9.2
npis m=3, kside=3, conlev=0.95
npis m=3, kside=2, conlev=0.95
c  #
c  #chapter 5
c  #
read 100 observations
 1.49  1.66  2.05  2.24  2.29  2.69  2.77  2.77  3.10  3.23
 3.28  3.29  3.31  3.36  3.84  4.04  4.09  4.13  4.14  4.16
 4.57  4.63  4.83  5.06  5.17  5.19  5.89  5.97  6.28  6.38
 6.51  6.53  6.54  6.55  6.83  7.08  7.28  7.53  7.54  7.68
 7.81  7.87  7.94  8.43  8.70  8.97  8.98  9.13  9.14  9.22
 9.24  9.30  9.44  9.69  9.86  9.99 11.28 11.37 12.03 12.32
12.93 13.03 13.09 13.43 13.58 13.70 14.17 14.36 14.96 15.89
16.57 16.60 16.85 17.18 17.46 17.74 18.40 18.78 19.84 20.45
20.89 22.28 22.48 23.66 24.33 24.72 25.46 25.67 25.77 26.64
28.28 28.28 29.07 29.16 31.14 31.83 33.24 37.32 53.43 58.11
c  #section 5.2.2
ciyp .5 kside=3, conlev=0.99
ciyp .5 kside=3, conlev=0.95
ciyp .1 kside=3, conlev=0.95
ciyp .9 kside=3, conlev=0.95
c  #section 5.2.4
ciyp p=.1, kside=1, conlev=0.95
ciyp p=.9, kside=2, conlev=0.95
epyp (100,5,101,.1)
epyp (100,0,95,.1)
c  #
c  #section 5.3.2
c  #
npti p=0.90, kside=3, conlev=0.95
npti p=0.99, kside=3, conlev=0.95
epti(100,1,100,0.99)
```

```
epti(473,1,473,0.99)
c  #
c  #section 5.3.4
c  #
epti(299,1,300,0.99)
npti p=0.9, kside=2, conlev=0.95
ciyp p=0.9, kside=2, conlev=0.95
npti p=0.9, kside=1, conlev=0.95
ciyp p=0.9, kside=1, conlev=0.95
npti p=0.1, kside=2, conlev=0.95
ciyp p=0.1, kside=2, conlev=0.95
npti p=0.1, kside=1, conlev=0.95
ciyp p=0.1, kside=1, conlev=0.95
c  #
c  #section 5.4.2
c  #
pikm  k=5, m=5, kside=3, conlev=0.90
pikm  k=4, m=5, kside=3, conlev=0.95
c  #
c  #table 5.2
c  #
epkm(100,1,100,5,3)
epkm(100,1,100,5,4)
epkm(100,1,100,5,5)
epkm(100,4,097,5,1)
epkm(100,4,097,5,2)
epkm(100,4,097,5,3)
epkm(100,4,097,5,4)
epkm(100,4,097,5,5)
epkm(100,10,091,5,1)
epkm(100,10,091,5,2)
epkm(100,10,091,5,3)
epkm(100,10,091,5,4)
epkm(100,10,091,5,5)
c  #
c  #
c  #section 5.4.4
c  #
pikm k=4, m=5, kside=1, conlev=0.99
```

```
pikm k=4, m=5, kside=2, conlev=0.99
pikm k=3 m=5 kside=1 conlev=0.99
pikm k=3 m=5 kside=2 conlev=0.99
pikm k=4 m=5 kside=3 conlev=0.99
epkm(100,1,100,5,4)
c  #
c  #table 5.3
c  #
epkm(100,1,101,5,3)
epkm(100,1,101,5,4)
epkm(100,1,101,5,5)
epkm(100,4,101,5,1)
epkm(100,4,101,5,2)
epkm(100,4,101,5,3)
epkm(100,4,101,5,4)
epkm(100,4,101,5,5)
epkm(100,10,101,5,1)
epkm(100,10,101,5,2)
epkm(100,10,101,5,3)
epkm(100,10,101,5,4)
epkm(100,10,101,5,5)
c  #
c  #table 5.2  upper
c  #
epkm(100,0,100,5,3)
epkm(100,0,100,5,4)
epkm(100,0,100,5,5)
epkm(100,0,097,5,1)
epkm(100,0,097,5,2)
epkm(100,0,097,5,3)
epkm(100,0,097,5,4)
epkm(100,0,097,5,5)
epkm(100,0,091,5,1)
epkm(100,0,091,5,2)
epkm(100,0,091,5,3)
epkm(100,0,091,5,4)
epkm(100,0,091,5,5)
c  #
c  #section 5.5.3
```

```
c  #
piyj j=20, m=39, kside=3, conlev=0.95
piyj j=30, m=59, kside=3, conlev=0.95
piyj j=30, m=59, kside=1, conlev=0.95
piyj j=30, m=59, kside=2, conlev=0.95
c  #
c  #chapter 6
c  #
bdata x=1, n=10
bcip kside=3, conlev=0.95
bcip kside=2, conlev=0.95
ndata xbar=8.05, sd=1.09, nnorm=10
npgt k=10, kside=3, conlev=0.95
npgt k=10 kside=2 conlev=0.95
bdata x=20, n=1000
bcip kside=3, conlev=0.95
bcip kside=2, conlev=0.95
c  #
c  #section 6.3.2
c  #
bple k=2, m=50, kside=3, conlev=0.95
bple k=2, m=50, kside=1, conlev=0.95
c  #
c  #section 6.4.3
c  #
btb  p=0.90, m=50, kside=2, conlev=0.95
c  #
c  #section 6.5.3
c  #
bpi  m=1000, kside=3, conlev=0.95
bpi  m=1000, kside=2, conlev=0.95
c  #
c  #chapter 7
c  #
pdata x=24 occurrences in n=5 units
c  #
c  #section 7.2.4
c  #
pcir kside=3, conlev=0.95
```

```
pcir kside=2, conlev=0.95
c  #
c  #section 7.3.2
c  #
pple k=5, m=0.5, kside=3, conlev=0.95
pple k=5, m=0.5, kside=1, conlev=0.95
c  #
c  #section 7.4.3
c  #
ptb p=0.90, m=1, kside=2, conlev=0.95
c  #
c  #section 7.5.4
c  #
ppi m=.5 kside=2, conlev=0.95
ppi m=4, kside=3, conlev=0.95
ppi m=3, kside=3, conlev=0.95
c   #
c  #section 8.2.2
c  #
read 2 observations
15.13 15.25
cndata to compute xbar and sigma
ncim kside=3 conlev=.95
ncim kside=3 conlev=0.99
bdata x=1 n=1
bcip kside=3 conlev=.95
bdata x=2 n=2
bcip kside=3 conlev=.95
c  #
c  #section 8.5.3
bdata 5 failures in 10 observations
bcip kside=3 covlev=0.90
bdata 15 failures in 30 observations
bcip kside=3 covlev=0.90
bdata 50 failures in 100 observations
bcip kside=3 covlev=0.90
bdata 100 failures in 200 observations
bcip kside=3 covlev=0.90
c  #section 8.6.2
```

```
pdata 80 flaws in 800 test units
pcir  kside=1 conlev=0.95
pcir  kside=2 conlev=0.95
c  #section 8.6.3
pdata 68 flaws in 667 test units
pcir  kside=1 conlev=0.95
pcir  kside=2 conlev=0.95
c
c    chapter 9
c  #
c  #section 9.2.3
c  #              normal dist 2-sided normal-theory tolerance interval
c  #
ndata xbar=0 s=1 n=91
nti  p=.9, kside=3, conlev=.95
c  #evaluate at p*=.96       to check delta=0.10
c  #
c  #section 9.2.4   1-sided lower normal theory tolerance bound
c  #
ndata xbar=0 s=1 n=369
nti  p=.99 kside=1, conlev=.95
c  #evaluate interval at p*=.997, look for delta=.01
c  #
c  #section 9.3.1  normal distribution demonstration
c  #
ndata xbar=0 s=1 n=138
nti p=.95 kside=2 conlev=.9
c  #evaluate at p*=.98 looking for pdem=.95
c  #
c  #section 9.4.2
c  #              minimum size two-sided d-f tolerance bound
c  #
nvec=46
npti    p=.9, kside=3 conlev=.95
epti 46 1 46 .9
epti 46 1 46 .96
c  #
c  #section 9.4.4
c  #              minimum size one-sided d-f tolerance bound
```

```
c  #
nvec=299
npti    p=.99, kside=1 conlev=.95
epti 299 1 300 .99
epti 299 1 300 .997
c  #
c  # section 9.5.1
c  #           controling a two-sided d-f tol int
c  #
nvec 154
npti p=.90 kside=3 conlev=.95
epti 154 5 150 .90
epti 154 5 150 .96
c  #
c  #section 9.5.2a  one sided sample size for d-f tol interval
c  #
nvec=1050
npti p=.99 kside=1 conlev=.95
epti 1050 6 1051 .99
c  #evaluate at p*=.997 looking for delta=.01
epti 1050 6 1051 .997
c  #
c  #section 9.5.2b  one sided sample size for d-f tol interval
c  #
nvec=1941
npti p=.99 kside=1 conlev=.95
epti 1941 13 1942 .99
c  #evaluate at p*=.997 looking for delta=.01
epti 1941 13 1942 .997
c  #
c  #section 9.6.1  demonstration for a proportion
c  #
nvec=531
npti p=.99 kside=1 conlev=.90
c  #evaluate at p*=.999 looking for delta=.05
npti p=.99 kside=2 conlev=.90
epti 531 0 529 .99
epti 531 3 532 .99
epti 531 0 529 .999
```

```
epti 531 3 532 .999
stop
c  #chapter 11
c  #
c  #section 13.2---example 1
c  #
read 6 observations
170.5 172.5 169.5 174.0 176.0 168.0
cndata
ndata xbar=171.8 s=3.0 n=6
bdata 6 conforming in 6 trials
c  #section 13.2.4
bcip kside=1 conlev=0.75
bcip kside=1 conlev=0.90
bcip kside=1 conlev=0.95
bcip kside=1 conlev=0.99
c  #section 13.2.5
nplt 180 kside=1 conlev=0.75
nplt 180 kside=1 conlev=0.90
nplt 180 kside=1 conlev=0.95
nplt 180 kside=1 conlev=0.99
c  #section 13.2.6
nti  p=.99 kside=2 conlev=0.90
nti  p=.99 kside=2 conlev=0.95
c  #section 13.2.7
ciyp  p=.99 kside=1 conlev=0.90
oot1 n=548  pval=0.99 conlev=.90
c  #
c  #section 13.3---------example 2
c  #
read 5
32 27 37 33 35
cndata
c  #section 13.3.3
npim m=7 obs kside=2 conlev=0.95
c  #section 13.3.4
pdata 164   in 5
ppi poisson prediction m=7 kside=2 conlev=0.95
c  #
```

```
c  #section 13.4----------example 3
c  #
read 20
       73.2  67.8  68.5  73.8  69.3
       70.9  65.4  71.2  72.4  69.6
       67.1  69.2  66.5  72.9  75.4
       74.2  69.1  64.0  68.9  70.2
cndata to compute xbar and s
bdata 19 conforming out of 20
c  #section 13.4.3
bcip kside=1 conlev=0.95
c  #section 13.4.4
nplt  75 kside=1, conlev=0.95
c  #section 13.4.5
pikm k=3 out of m=3 kside=2 conlev=0.95
epkm n=20 ir=0 is=20 m=3 k=3
npik k=3 out of m=3 kside=2 conlev=0.95
c  #
c  #section 13.5----------example 4---------sample size problem
c  #
c  #
c  #section 13.6---------example 5
c  #
read 27 observations
   48. 94. 112. 44. 93.
  198. 43. 52. 35. 170.
   25. 22. 44. 16. 139.
   92. 26. 116. 91. 113.
   14. 50. 75. 66. 43.
   10. 83.
c  #section 13.6.5
bdata 0 in 27 trials
bcip kside=2 conlev=0.95
ndata xbar=4.01 s=0.773 n=27
npgt  x=5.70 kside=2 conlev=0.95
cndata
npgt x=300 kside=2 conlev=0.95
c  #
c  #section 13.7-----------example 6
```

```
c  #
bdata 2 out of 1000 failed
c  #section 13.7.2
bcip kside=3 conlev=0.90
bpgt y=0 in 110 observations kside=3 conlev=0.90
c  #
c  #section 8-------------example 7
c  #
read 16 observations
  552.  586.  702.  722.
  742.  790.  800.  838.
  838.  921.  960.  981.
  994. 1035. 1110. 1405.
cndata
bdata 16 out of 16 pass
bcip kside=1 conlev=0.90
bpge x=32 m=32 kside=1 conlev=0.90
npgt 450 kside=1 conlev=0.90
npgt 450 kside=3 conlev=0.90
stop
```

D Warning And Error Messages

An error checking facility has been written into **StInt**. **StInt** detects many errors while reading commands. Other errors can only be detected at some later time. There may be some errors that **StInt** will not detect. Users should scrutinize output carefully to be sure that results seem plausible. The following describes some of the errors that **StInt** will detect.

- Unrecognized commands (usually caused by a typing error) are flagged and ignored, after which **StInt** continues to the next command.
- If the required number of arguments is not present a message will be printed, and **StInt** will continue to the next command.
- If an argument is out of its allowable range, an error message is printed giving arguement, the upper and lower values of the allowable range, and a substitute value. **StInt** then uses the substitute value and continues to the next command.

In an effort to reduce computer storage requirements and because most other error conditions should occur infrequently, many abnormal conditions which are detected by **StInt** are followed by a printed message and an error number. A detailed explanation of the condition which caused a particular error can be found in the following list.

It is possible that some error conditions will not be detected by **StInt** but will generate errors which are detected by the operating system of the user's computer.

ERROR MESSAGES

- 991 Incorrect command sequence.
- 992 Incorrect number of arguments on the previous command.
- 993 Required data not in place.
- 994 Error in input.

Other error numbers have 4 digits. The first three digits refer to the internal "command number" and the last digit indicates the argument number on the command that triggered the error.

References

David, H. A. (1981), *Order Statistics,* (Second Edition) New York: John Wiley & Sons, Inc.

Hahn, G. J., and Meeker, W.Q. (1991), *Statistical Intervals: A Guide for Practitioners*, New York: John Wiley & Sons, Inc.

Odeh, R. E., and Owen, D. B. (1980), *Tables for Normal Tolerance Limits, Sampling Plans, and Screening,* New York: Marcel Dekker, Inc.

Odeh, R. E., and Owen, D. B. (1983), *Attribute Sampling Plans, Tables of Tests and Confidence Limits for Proportions,* New York: Marcel Dekker, Inc.