

Chapter 8

Sample Size Program

8.1 Introduction

The success of a survival study can be measured, in part, by the level of certainty associated with the estimated survival probabilities. The greater the level of sampling precision, the greater the certainty the estimate reflects the true but unknown survival rate. Variance estimates (i.e., $\hat{var}(\hat{S}|S)$) and standard errors (i.e., $\sqrt{\hat{var}(\hat{S}|S)} = \hat{SE}(\hat{S}|S)$) both express the measurement errors associated with estimating the survival rate of a specific release group in time and space.

The measurement error associated with a specific survival estimate will be a function of (a) actual survival probabilities, (b) capture probabilities, (c) release sizes of marked animals, and (d) rate marked animals are knowingly removed from the investigation. Anticipated measurement error for a survival study can be projected by having preliminary “guesstimates” of these quantities. Furthermore, investigators can determine required release sizes and capture probabilities to achieve a pre-specified level of sampling precision. This chapter explains how a utility associated with SURPH.1 called SAMPLE-SIZE can be used to investigate sample sizes and precision of release-recapture and known-fate studies whose focus is to estimate period-specific survival probabilities.

Neither the survival probabilities nor the capture probabilities will be known prior to beginning the study. As such, the user must make rough guesses as to the values of these quantities in sample size calculations. One suggested way to improve on these guesses is to use a pilot study (Skalski 1993). Because the survival and capture probabilities are often beyond the control of the experimenter, the primary tool the experimenter has for changing the precision is to alter the release size. The Sample-Size program can be used to investigate the interrelationships between release size, survival probability, capture probability, and known removals on the precision of the survival estimates,

8.2 Statistical Definition of Precision

Common measures of precision include the standard error of a parameter estimate (i.e., $\hat{SE}(\theta) = \sqrt{\widehat{Var}(\hat{\theta}|\theta)}$) and the coefficient of variation (i.e., $CV = \sqrt{\widehat{Var}(\hat{\theta}|\theta)} / \hat{\theta}$). The standard error expresses precision in absolute terms while the coefficient of variation expresses precision in relative terms. A third approach expresses precision in a probabilistic manner reflecting the fact that the parameter estimate is a random variable. In estimating survival probabilities, a useful measure of precision is expressed as

$$P(|\hat{S} - S| < \epsilon) \leq 1 - \alpha, \quad (8.1)$$

the probability that the absolute deviance between the estimate and true value (i.e., $|\hat{S} - S|$) is arbitrarily small (i.e., $< \epsilon$) with some high probability of occurrence (i.e., $1 - \alpha$). An alternative measure of precision,

$$P\left(\left|\frac{\hat{S} - S}{S}\right| < \epsilon\right) = 1 - \alpha \quad (8.2)$$

expresses the deviance in relative terms (i.e., $(\hat{S} - S) / S$). Expression (8.2), while a valid expression of precision, is less likely to be used in estimating survival because survival is already a relative measure from 0-1.

For example, if there was a desire for a study to estimate survival (S) within ± 0.10 , 80 percent of the time, expression (8.1) would be rewritten as

$$P(|\hat{S} - S| < 0.10) = 0.80$$

where $\epsilon = 0.10$ and $\alpha = 0.20$. The demands on precision increase as ϵ and/or α become smaller. An investigator must choose the most appropriate level of precision for the goals of the study. Often the investigator may want to evaluate alternative levels of study effort and the associated changes in precision that may be anticipated.

precision (i.e., expression 8.1) can also be interpreted where ϵ is the half-width of an anticipated $(1 - \alpha)$ 100% confidence interval estimate of \hat{S} . Values of ϵ and α can therefore be selected to achieve the desired width (a) of a confidence interval at the a significance level. The relationship **between** the precision expression (8.1) and a $(1 - \alpha)$ 100% confidence interval can be seen from the following manipulations where

$$P(|\hat{S} - S| < \epsilon) = 1 - \alpha$$

$$P(-\epsilon < \hat{S} - S < \epsilon) = 1 - \alpha$$

$$P\left(\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} < \frac{\hat{S} - S}{\sqrt{\text{Var}(\hat{S}|S)}} < \frac{\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = 1 - \alpha$$

and assuming S normally distributed

$$P\left(\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} < Z < \frac{\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = 1 - \alpha$$

$$1 - 2P\left(Z < \frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = 1 - \alpha$$

$$P\left(Z < \frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = \frac{\alpha}{2}$$

$$\Phi\left(\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = \frac{\alpha}{2}$$

$$\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} = Z_{\frac{\alpha}{2}}$$

$$\frac{\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} = Z_{1-\frac{\alpha}{2}}$$

$$\epsilon = Z_{1-\frac{\alpha}{2}} \sqrt{\text{Var}(\hat{S}|S)}$$

where Z is a standard normal deviant (i.e., $P(|Z| > Z_{1-\frac{\alpha}{2}}) = \alpha$).

In the `Sample_Size` program, two measures of precision will be presented. The two measures are the anticipated standard error (i.e., $\sqrt{Var(\hat{S}|S)}$) and values of α and $1 - \alpha$ from probabilistic expression (8.1).

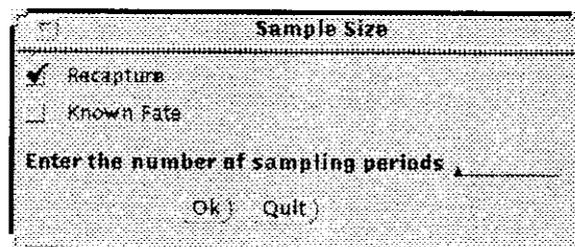
8.3 Using the Sample-Size Program in Designing a Single-Population Study

This section illustrates the main features of the `Sample_Size` program.

To begin the Sample Size program, at the prompt, type “sample”:

> sample

Left-click on either the Recapture or the Known-Fate Check-Box. The Recapture Check-Box should be selected if the study that will be used is a release-recapture study. Select the Known-Fate box if the study is a telemetry study. Enter the number of sampling occasions in the blank field, then select OK if the information is correct. The number of sampling occasions is equal to the number of times the user will attempt to detect marked animals. Thus, to calculate the number of sampling occasions, do not include the initial release. If something is incorrect, go back and change the selection before selecting OK. Select Quit to leave the Sample Size program.



In a release-recapture study, both survival and capture probabilities are unknown random variables, so the user will be asked to provide “guesstimates” of both survival and capture probabilities during each period. In a known-fate study, only the survival probabilities are random variables (the probability of capture in a known-fate study is 1.00), therefore the user will be asked to provide estimates of only survival probabilities during each period.

After you have checked the appropriate study and entered the number of sampling periods, left-click on OK. The next window that will appear is the Sample Input Window. The Sample Input Window is used to iteratively investigate the effects of changing sample size, survival and recapture probabilities.

Sample Input					
	S1:	S2:	S3:	S4:	S5:
R0:	R1:	R2:	R3:	R4:	
	P1:	P2:	P3:	P4:	P5:
	D1:	D2:	D3:	D4:	Ok

The Sample Input Window is the heart of the Sample Size program. It is here that the user will alter the release sizes and possibly the capture probabilities to arrive at the level of effort needed to get the precision necessary for the study. Each input has a letter-number designation. The letter designates the type of value; the number designates the sampling occasion. Letter designations are S (for Survival Probability), R (for Release Number), P (for Capture Probability), and D (for Proportion Removed/Known Dead). The R_j field should be used to enter the number of unmarked animals that will be released on the i th occasion. The S_j field should be used to enter the estimated survival probability of the population between sampling occasions “ $j-1$ ” and “ j ”. The P_j field should be used to enter the estimated capture probability of the population on sampling occasion “ j ”. And similarly, the D_j field should be used to enter the estimated proportion of the tagged animals that are captured that will be removed on sampling occasion “ j ”. This removal can be due to either accidental deaths associated with the capture process or due to purposeful removals.

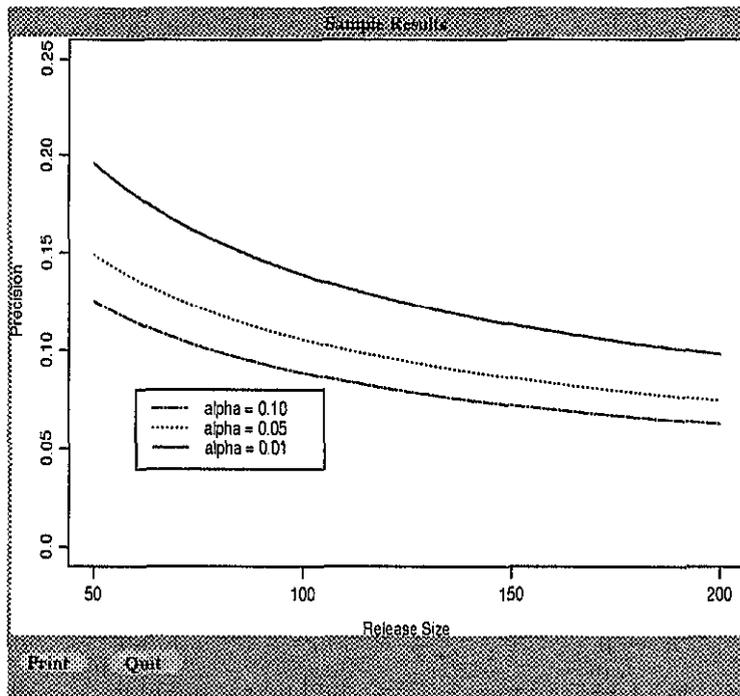
Fill in the blank fields with guesstimates of survival probabilities, capture probabilities, and number released. All fields must be filled before the user can proceed. Once all the blank fields have been filled, left-click on the OK.

The output windows are a table of estimated standard errors for the survival estimates and a graphical display relating precision (defined as half the width of a $1 - \alpha$ confidence interval around the survival estimate) to the release size and the α -level. The estimates of the standard errors are computed using the closed form equations for the variance estimates based on the multinomial likelihood (Cormack 1964).

Two windows will appear, a text window and a graphics window. The text window displays the per-period survival, and the associated projected standard error.

Period	Survival Probability	Expected Std. Error
1	0.950	0.042
2	0.950	0.031
3	0.950	0.025
4	0.950	0.029

The graphics window displays the precision (defined as the half-width of the α -level confidence interval) as a function of the release size. There are three curves in the graphics window, associated with three α -levels (0.01, 0.05, and 0.10).



Sample size determination *now* becomes an iterative process. If the current release size(s) does not give sufficient precision, the user can either find a way to increase the capture probability(ies), or the user can increase the release size(s). Generally, in order to half the estimated standard error, the user must quadruple the release size. If there is low survival, the user may get better precision by going from a single-release to a multiple-release. This way, a

higher sample size is maintained throughout the study. If survival is high throughout the study, there is little to be gained by using multiple releases.

Example 1 - Release size to achieve a desired precision level

Suppose that in a particular survival study, the desired precision of the survival estimate is to be within ± 0.05 of the true value 95% of the time ($\epsilon = 0.05$, $\alpha = 0.05$). The study will consist of a single release followed by 4 survey periods. The anticipated survival during any one period will be approximately 0.70, and the probability of capture will be about 0.40. No removals of tagged animals are planned. The question that can then be asked is how many animals must be released in order to achieve the desired level of precision.

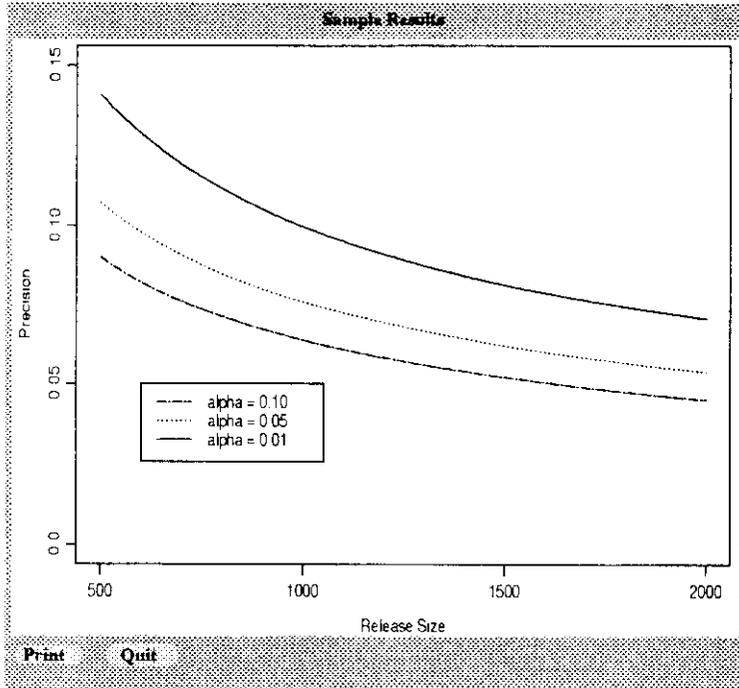
Begin the Sample-Size program. Check the Recapture check-box, and enter “4” in the Number of Sampling Periods field. Enter the approximate survival, capture and removal rates into the blank fields of the Sample Input Window. Once completed, the Sample Input Window should look like the window below.

Sample Input				
	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
RB: 1000	R1: 0	R2: 0	R3: 0	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	Ok

Left-click on OK when all blanks have been filled.

Initially, we guess that 1000 animals maybe sufficient. However, when we examine the anticipated precision curve at $\alpha = 0.05$, we discover that a release size of 1000 will yield a precision of about ± 0.08 (Figure 8.1 (A)). Using the “rule-of-thumb” discussed previously, the subsequent guesstimate of R_1 was quadrupled to a release size of 4000 (Figure 8.1 (B)), in an attempt to produce the desired level of precision in the first period. However, notice that although the precision is now within the desired range for the first period survival, the precision for both the second and third period survivals is outside the range desired (Figure 8.1 (C)). If we wish to maintain the precision desired throughout the study, we will either need to increase the initial release, or supplement the number of marked animals during subsequent releases.

(A)



(B)

Sample Input				
	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
R0: 4000	R1: 0	R2: 0	R3: 0	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	OK

(C)

Sample Results		
Print	Quit	
Period	Survival Probability	Expected Std. Error
1	0.700	0.019
2	0.700	0.031
3	0.700	0.045

Figure 8.1 A selection of the windows from the first single-release example. (A) Graphical display of the initial guess (i.e., $R_0 = 1000$). (B) Sample Input Window for the intermediary guess (i.e., $R_0 = 4000$). (C) Textual output of the intermediate guess (i.e., $R_0 = 4,000$).

If we try to achieve the desired precision throughout the study by increasing the initial release, trial and error indicates that the initial release size must be near 14,000 animals. This release size drastically increases the precision for the first and second period survival estimates, but just achieves the precision desired during the third period (Figures 8.2 (A) and (B)).

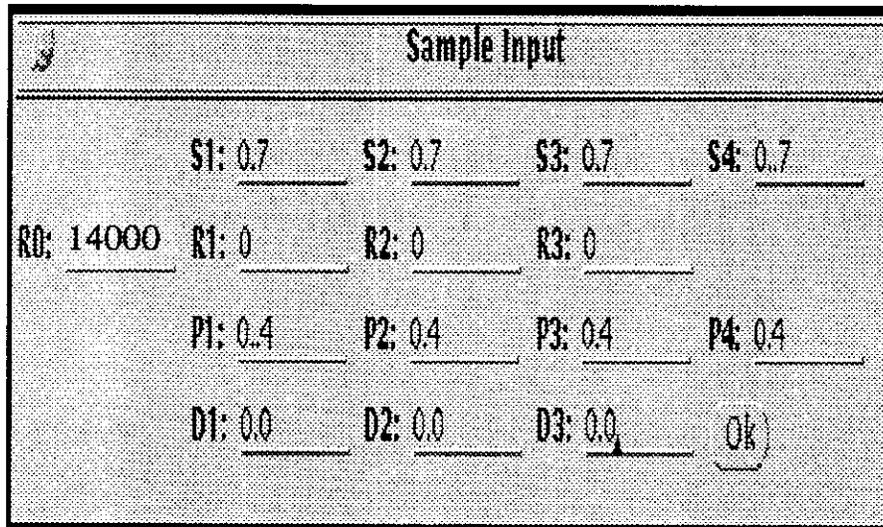
It may be useful to see what period-specific release sizes would be necessary to achieve the desired precision for each survival estimate. Using the information contained in Figure 8.1(A), use 2500 animals in the initial release. This gives the desired precision for the first period. Then begin adding animals to subsequent releases until the standard error for each of the per period survival estimates is below $\epsilon = 0.05$ (this is because at $\alpha = 0.05$, the half-width of the confidence interval is anticipated to be $1.96 * (\text{estimated standard error})$). Iterative trial and error indicates that releases of 1400, 1400, 1700 and 1700 would achieve the desired precision throughout the study (Figures 8.3 (A) and (B)). We conclude that a total of 6200 animals must be released under the multiple-release study design, whereas 14,000 animals were necessary under the single-release design. Thus, if the multiple-release study design is an option, less than half as many animals would have to be marked under the multiple-release study design compared to the single-release design to achieve the desired precision throughout the study.

Example 2 - Precision at a specified release size

In the previous example, the precision was pre-specified and the release sizes were determined to achieve the desired precision level. For many studies, however, the release sizes are fixed, either because of budgetary constraints or because of time and effort constraints. Therefore, a common issue is determining the level of precision likely to be achieved with a predetermined level of effort.

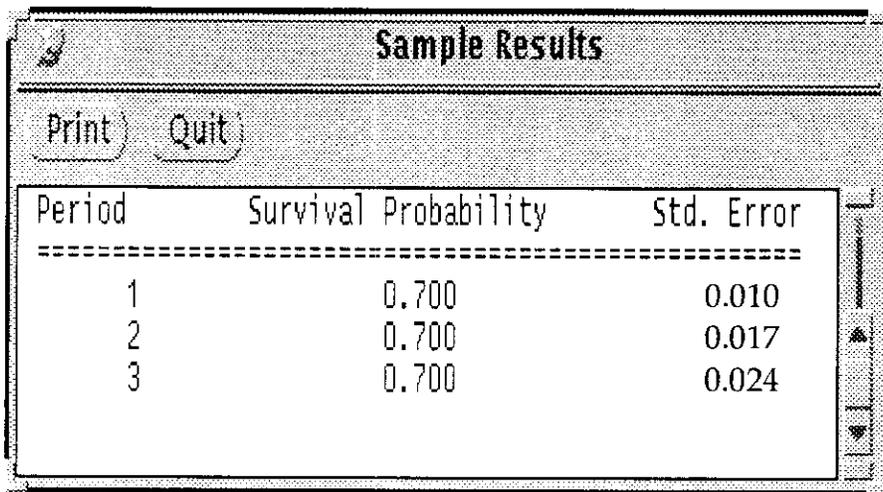
Assume that only an initial release of 14,000 marked animals is feasible. The study will be 4 periods long. The estimated survival during any one period will be approximately 0.70, and that the probability of capture will be about 0.40. No removals are anticipated. The question is, at the $\alpha = 0.05$ level, what is the anticipated level of precision (i.e., half-width of a 95% CI) that can be achieved with this level of effort.

(A)



A graphical window titled "Sample Input" with a standard Mac OS-style title bar. The window contains several input fields arranged in a grid. The fields are labeled as follows: S1: 0.7, S2: 0.7, S3: 0.7, S4: 0.7; R0: 14000, R1: 0, R2: 0, R3: 0; P1: 0.4, P2: 0.4, P3: 0.4, P4: 0.4; D1: 0.0, D2: 0.0, D3: 0.0. There is an "Ok" button in the bottom right corner.

(B)



A graphical window titled "Sample Results" with a standard Mac OS-style title bar. At the top, there are two buttons: "Print" and "Quit". Below the buttons is a table with three columns: "Period", "Survival Probability", and "Std. Error". The table is separated from the rest of the window by a dashed line. The data in the table is as follows:

Period	Survival Probability	Std. Error
1	0.700	0.010
2	0.700	0.017
3	0.700	0.024

Figure 8.2 A selection of the windows from the first single-release example. (A) Graphical display of the Sample Input Window for the final guess (i.e., $R_0 = 14,000$) using a single-release design. (B) Textual output of the Sample Results Window for the final guess (i.e., $R_0 = 14,000$) using a single-release design.

(A)

Sample Input				
	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
RD: 1400	RI: 1400	R2: 1700	R3: 1700	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	Ok

(B)

Sample Results		
Print	Quit	
Period	Survival Probability	Std. Error
1	0.700	0.025
2	0.700	0.023
3	0.700	0.025

Figure 8.3 A selection of the windows from a multiple-release example. (A) The Sample Input Window for the Fixed Precision Example, with the final release sizes under a multiple-release design. (B) Textual output of the Sample Results Window for the Fixed Precision Example using a multiple-release design.

As before, fill in the blanks of the Sample Input Window with the known information (Figure 8.2 (A)). When finished, select OK.

Recall that at $\alpha = 0.05$, the half-width of the confidence interval will be approximately 1.96 times the estimated standard error. Thus, using this approximation, we find that the precision for the first period survival is anticipated to be $\epsilon = 0.022 (-1.96SE(\hat{S}))$, the precision for the second period survival is anticipated to be $\epsilon = 0.033$, and the precision for the third period survival is anticipated to be $\epsilon = 0.049$ (Figure 8.2 (B)).

Assume that an alternative study design allows multiple releases of 1500 newly marked animals per period. Using the same survival and capture probabilities as before, at the $\alpha = 0.05$ level, what level of precision will be achieved?

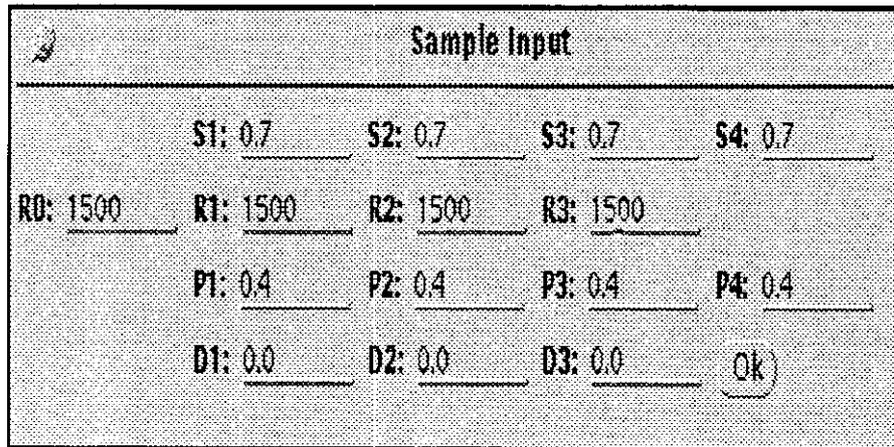
Fill in the blanks of the Input Window with the known information (Figure 8.4 (A)). When finished, select OK.

Because we expect the parameter estimates to be distributed asymptotically-normal, the half-width of the confidence interval will be 1.96 times the estimated standard error. Using this fact, we find that the precision for the first period survival is anticipated to be $\epsilon = 0.047$, the precision for the second period survival is anticipated to be $\epsilon = 0.045$, and the precision for the first period survival is anticipated to be $\epsilon = 0.051$ (Figure 8.4 (B)).

8.4 Conclusion

The Sample Size program can be used to determine the level of effort needed to achieve a desired level of precision for \hat{S} or the precision that is attainable given a prescribed level of effort. As such, this tool should be used during the design phase of the survival study, and can be used in concert with the SURPH analysis of survival data.

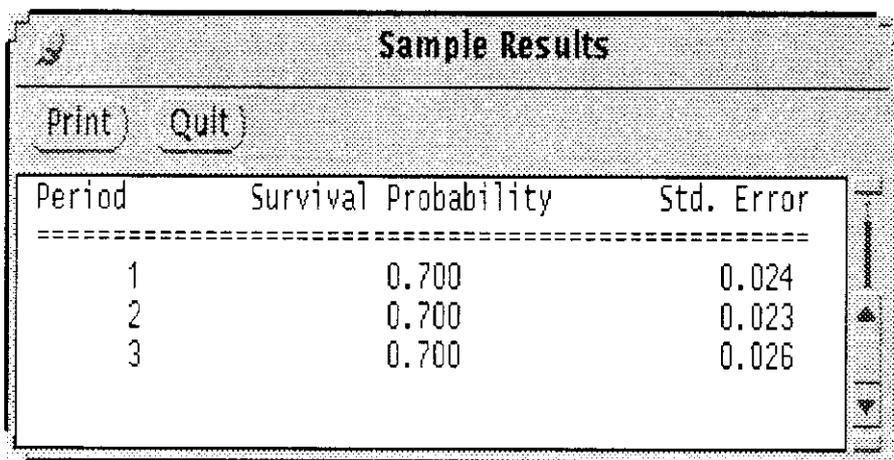
(A)



The 'Sample Input' window contains the following fields:

	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
RD: 1500	R1: 1500	R2: 1500	R3: 1500	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	Ok

(B)



The 'Sample Results' window displays the following data:

Period	Survival Probability	Std. Error
1	0.700	0.024
2	0.700	0.023
3	0.700	0.026

Figure 8.4 A selection of the windows from the second multiple-release example. (A) The Sample Input Window for Fixed Effort Example. (B) Textual output of the Sample Results Window for the Fixed Effort Example using a multiple-release design.

The examples in this chapter dealt only with estimating the precision of survival estimates from the study of a single marked cohort. In many studies, the desire is to determine how survival rates differ between populations or are affected by environmental or human influences. Readers are encouraged to read Skalski and **Robson** (1992) and **Burnham** et al. (1987) before designing a multi-population study.

Chapter 9

Data Tools Program

9.1 Introduction

The original release of SURPH.1 (Release date: December 1994) incorporated the ability to transform data (Data -> Data Transformations) within SURPH-readable datafiles. However, the Data Transformations utility had limited capabilities. Thus, we have created a more flexible and powerful database manager called **DataTool**. **DataTool** takes a SURPH-readable datafile, and creates a new SURPH-readable **datafile** that incorporates data manipulations of group and/or individual **covariates**. **DataTool** also allows the user to combine periods and populations within a datafile. This powerful tool should facilitate manipulations of the datafiles used in SURPH.1.

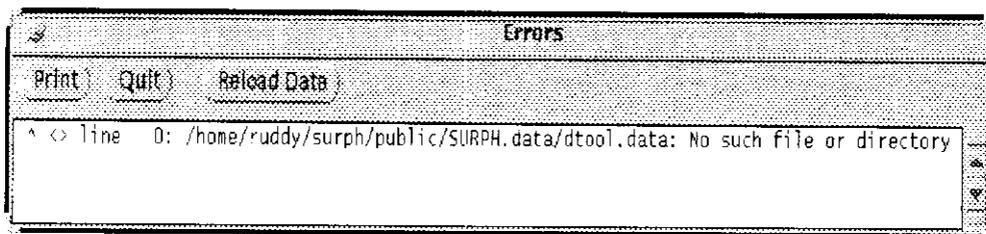
9.2 Description of the Tools Available in DataTool

This section will illustrate and briefly describe each window that the user will encounter while using **DataTool**. For more information on the use of **DataTool**, see Section 9.3.

To begin DataTools, at the prompt, type "datatool filename", where filename is the name of a SURPH-readable datafile. For the purposes of illustration, we will use the datafile "dtool.dat"

```
> datatool dtool.dat
```

If you attempt to access a file that is either not SURPH-readable or does not exist, the following dialog-box will appear.

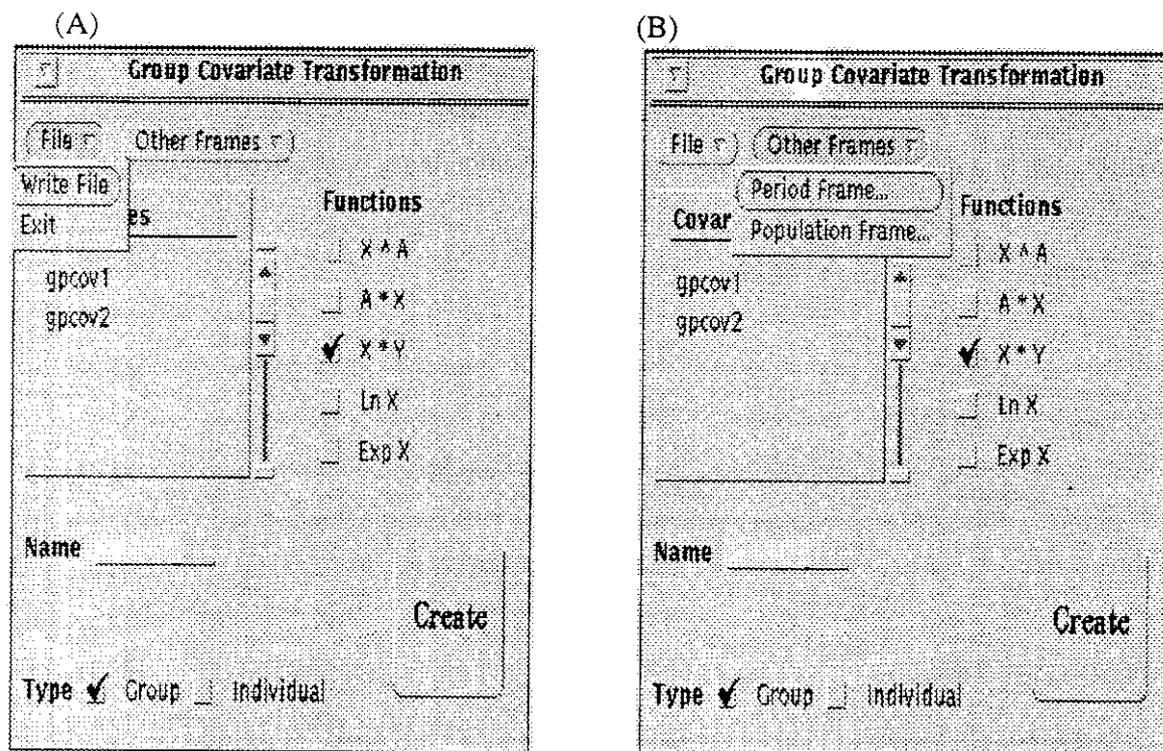


In the above example, the user attempted to access the file "dtool.data". An Error Window informs the user that the file "dtool.data" does not exist.

9.2.1 Group-Covariate Transformations

The initial window is the Group Covariate Transformation Window. The Group Covariate Transformation Window is used to generate new Group Covariates that are functions of the currently defined Group Covariates. As an example, suppose a group covariate in the data set was the concentration of ATPase, denoted “ATPase”. The user may wish to create a new covariate which is the natural log of the covariate “ATPase”. To create the new group covariate, the user could use the Group Covariate Transformation frame of **DataTool**.

On the left-hand side of the window, the group covariates are listed by name. On the right-hand side of the window, the transformations are listed. The (A) File Button at the top of the window allows the user to Write the transformed data to a new SURPH-readable file and to Quit Data Tool. The (B) Other Frames Button allows the user to move to the other types of transformations that are available (i.e., period and population transformations).

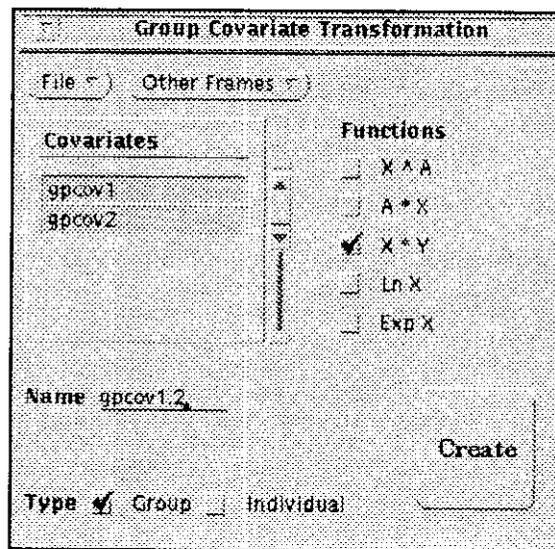


To create new data series that are a transformation of current data, select the covariate(s) from the Covariates List and the functional transformation from the Functions List. To name the

new Covariate, type the name you desire in the blank line following “Name” prior to pushing the Create Button. If you forget to include a name for the new variable, a pop-up window will appear and advise you to name the new variable. You cannot create a new variable without naming it. To create the new Covariate, place the arrow key above the Create Button, then left-click.

Available functional manipulations are:

- X^A** Raise the value of the selected Covariate (X) to the power “A”.
- A*X** Multiply the value of the selected Covariate (X) by “A”.
- X*Y** Multiply the value of one selected Covariate (X) by the value of another selected Covariate (Y).
- Ln (X)** Take the Natural Logarithm of the value of the selected Covariate (X).
- Exp (X)** Exponentiate the value of the selected Covariate (X).

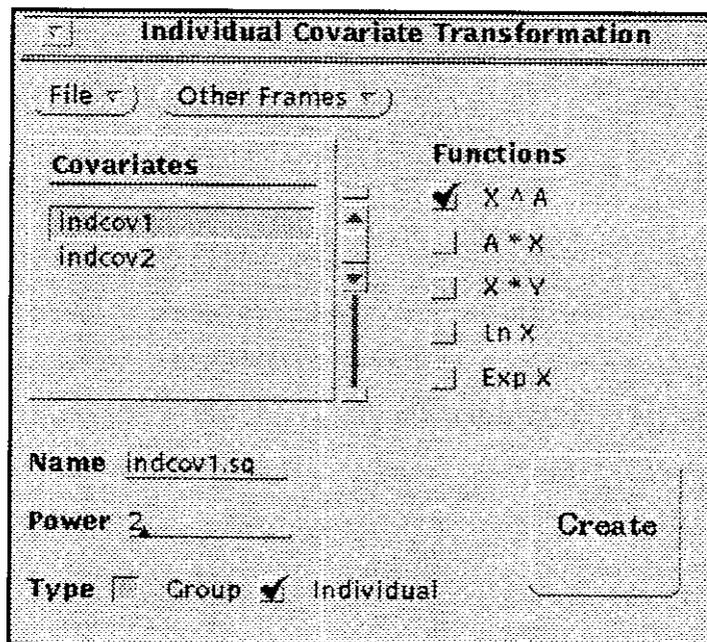


In the example window, left-clicking on the Create Button will create a new variable “gpcov1.2”. The values of gpcov1.2 for each population will be equal to the values of gpcov1 multiplied by gpcov2.

9.2.2 Individual-Covariate Transformations

The Individual Covariate Transformations allows the user to combine or alter the values of individual covariates that were collected. For example, suppose the user collected data on length, width and height of an object. If the object of interest is shaped like a cube, a variable of interest may be “Volume”, which could be approximated using “length x width x height”. To create the new covariate, the user could use the Individual-Covariate Transformation portion of DataTool.

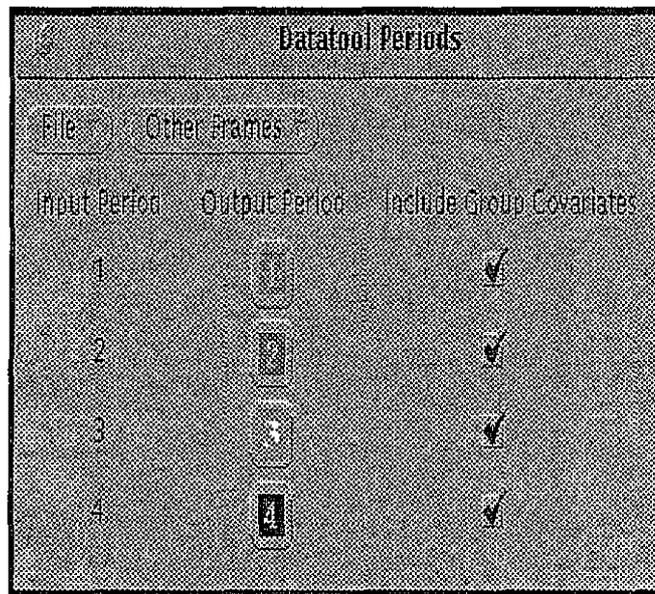
To get to the Individual Covariate Transform Window, left-click on the Individual checkbox at the bottom of the Group Covariate Transform Window. The Individual Covariate Transform Window looks similar to and operates identically to the Group Covariate Transform Window. The sole difference between the two windows is that the objects of the transformations are the individual covariates rather than the group covariates.



In the example above, the values of *indcov1* will be transformed using the transformation function X^A , where “X” is the covariate that will be used (i.e., *indcov1*), and “A” is the power to which that covariate will be raised (i.e., 2). The name of the new variable will be *indcov1.sq*, and the values of the new variable will equal $(indcov1)^2$.

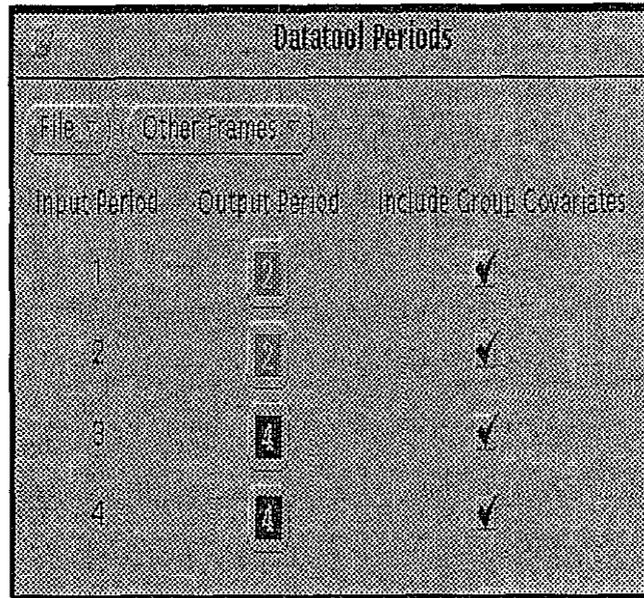
9.2.3 Period Transformations

The Periods Transformations Frame allows the user to combine periods over which the data was collected. For example, suppose the user collected data on a biweekly time-scale, but decided to analyze the data on a monthly time-scale. One way to collapse the data would be to assume that **all** animals that were released within a given month could be considered a single release. Similarly, all animals that were recaptured within a given month could be considered a single recapture. To collapse data in this manner, the user can use the Data Tool Periods Frame (Other Frames -> Period Frame).



On the far left of the Datatool Periods window are the values of the Input Period Values. To the right of the Input Period Values are a set of buttons that may be changed to collapse the current periods into new period designations. The algorithm uses the logical "OR" to collapse periods. To the far right of the Datatool Periods window is a column of checkboxes. A checked box indicates that the collapsed period should include the group covariates. If the covariate is time-invariant, it will not matter whether the boxes are checked or not. If the covariate varies through time and the checkboxes are checked, Data Tool alters the values of the current group covariate to the average of the group covariate values for all those periods that are combined. If

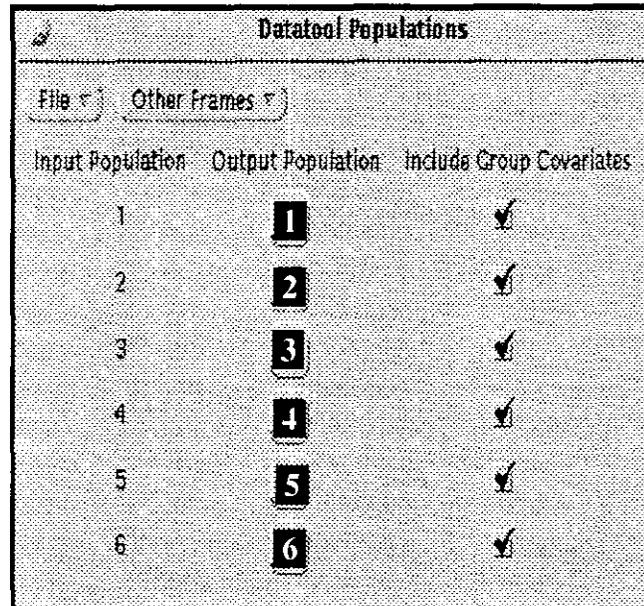
the **covariate** varies through time and the checkboxes are not checked, **DataTool** inserts zeros for all the group covariate values.



In the example above, periods 1 and 2 in the original dataset will be collapsed into a single period. Likewise, periods 3 and 4 in the original dataset will be collapsed into a single period. The checkboxes indicate that the new data will inherit the average values of the group covariates.

9.2.4 Population Transformations

The Population Transformations Frame allows the user to combine populations over which the data were collected. For example, suppose the user collected data from 2 blocks each within several different areas. One way to collapse the data would be to assume that both blocks within an area had the same survival, and thus constituted a single population, rather than two separate populations. To collapse data in this manner, the user can use the Data Tool Populations Frame (Other Frames -> Population Frame).



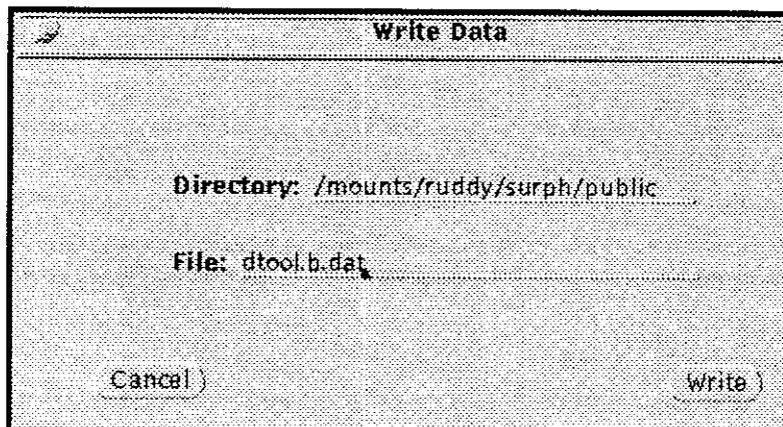
On the far left of the Data tool Populations window are the values of the Input Population Values. To the right of the Input Population Values are a set of buttons which may be changed to collapse the current populations into new population designations. The algorithm uses the logical “OR” to collapse populations. To the far right of the Datatool Populations window is a column of checkboxes. A checked box indicates that the collapsed population should include the group covariates. If the covariate is time-invsriate, it will not matter whether the boxes are checked. If the covariate varies through time and the checkboxes are checked, DataTool alters the values of the current group covariate to the average of the group covsriate values for all those populations that are combined. If the covsriate varies through time and the checkboxes are not checked, DataTool inserts zeros for all the group covariate values.

Input Population	Output Population	Include Group Covariates
1	2	<input checked="" type="checkbox"/>
2	2	<input checked="" type="checkbox"/>
3	4	<input checked="" type="checkbox"/>
4	4	<input checked="" type="checkbox"/>
5	6	<input checked="" type="checkbox"/>
6	6	<input checked="" type="checkbox"/>

In the example above, populations 1 and 2 in the original dataset will be collapsed into a single population. Likewise, populations 3 and 4 in the original dataset will be collapsed into a second population, and populations 5 and 6 will be collapsed into a third population. The checkboxes indicate that the new data will inherit the average values of the group covariates.

9.2.5 Writing to a File

Once the transformations have been completed, the file must be saved before it can be used in SURPH. To save a file, use the command sequence File -> Write File. A pop-up window will appear that will allow the user to save the transformed dataset to a new name.



To specify the directory where the datafile should be saved, left-click on the Directory field and type in the directory path. To specify the name of the new datafile, left-click on the File field, and type the name of the datafile. Once the correct directory has been specified and the file name has been entered into the field, left-click on the Write Button. In this example, the model file “dtool.b.dat” would be created in the directory “/mounts/ruddy/surph/public”.

9.3 Examples of Using DataTool

This section will be used to illustrate the concepts that were presented in Section 9.2. A dataset specifically created for this section (i.e., **dtool.dat**) will be used to illustrate the various procedures and pit-falls.

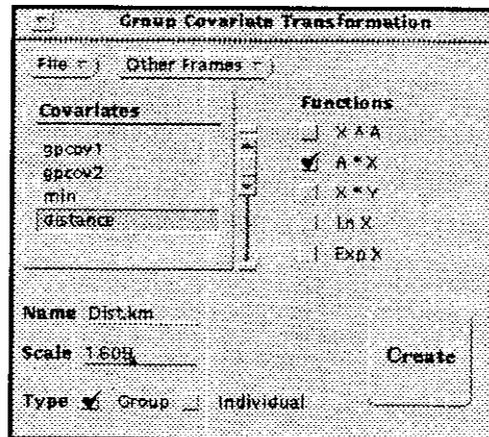
9.3.1 Group-Covariate Transforms

Suppose that the user originally collected data on the miles covered by an group of animals during a yearly migration (i.e., “distance”) and expected that variable to be related to survival. However, after the SLJRP-readable datafile had been constructed, the user decided that (s)he wanted to use kilometers as the independent variable instead of miles.

To begin DataTools, at the prompt, type “datatool filename”, where filename is the name of a SURPH-readable datafile. For the purposes of illustration, this example will use the datafile “dtool.dat”

```
> datatool dtool.dat
```

To create a new group-covariate variable that has values equal to a scalar of the original groupcovariate, select the current group-covariate variable from the list (i.e., “distance”). Check the A*X transform, and name the new variable “Dist.Km”. Put the conversion factor for miles to kilometers (i.e., 1.609) in the scale field. Left-click on Create. Save the new datafile under dtool.b.dat (File-> Write).



The new datafile now has a new group covariate called “Dist.Km” (Figure 9.1). Otherwise, except for some minor cosmetic differences, the new datafile is exactly like the old datafile.

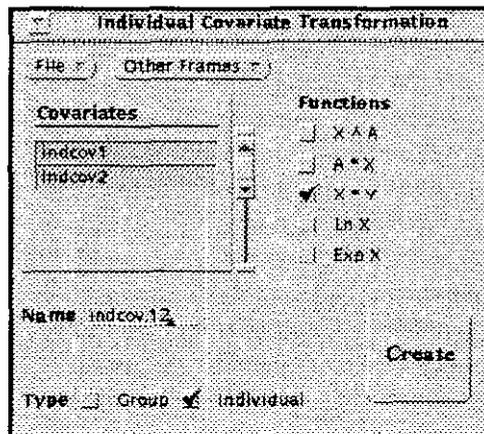
9.3.2 Individual-Covariate Transforms

Suppose that the user originally collected data on the length and girth of an animal. However, after the SURPH-readable datafile had been constructed, the user decided that (s)he wanted to use a combined “length x girth” measure as the independent variable instead of either the length or girth alone.

Begin DataTools. Check the box labelled Individual at the bottom of the Group Covariate Transformation Window. To create a new individual-covariate variable that has values equal to the product of two other individual covariates, select the current individual-covariate variables from the list (i.e., “indcov1” and “indcov2”). Check the X*Y transform, and name the new variable “indcov.12”. Left-click on Create. Save the new datafile under “dtool.c.dat”.

(A)	(B)
Data Simulated telemetry data from telem_sim_win	Data Simulated telemetry data from telem_sim_win
npop 6 nper 4	number-of-populations 6 number-of-periods 4
gcv name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000	group-covariate name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
gcv name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000	group_covariate name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000
gcv name min time 1 19 4 13 -7 19 4 13 -7	group-covariate name min time 1 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000
gcv name distance time 0 116 176 235 345 467 748	group-covariate name distance time 0 116.000000 176.000000 235.000000 345.000000 467.000000 748.000000
icov indcov1 icov indcov2	group-covariate name Dist.km time 0 166.644000 283.184000 378.115000 555.105000 751.403000 1203.532000
ntag 50 30 0 0 50 30 0 0 50 30 0 0	

Figure 9.1 Comparison of the (A) original datafile (Le., dtool.dat) to the (B) transformed datafile (i.e., dtool.b.dat).



The new datafile now has a new individual covariate called “indcov.12” (Figure 9.2) which is the product of the values of “indcov1” and “indcov2” for each individual (e.g., for the first individual, “indcov1” = 32.023, “indcov2” = **4.214**, and “indcov.12” = $32.023 \times 4.214 = 134.954$). Otherwise, except for some minor cosmetic differences, the new datafile is exactly like the old datafile.

9.3.3 Period Transforms

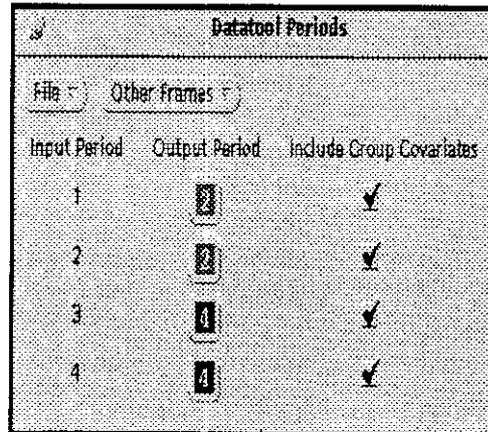
Suppose that the user originally collected data on a biweekly time-scale. However, after the SURPH-readable datafile had been constructed, the user decided that (s)he wanted to use a monthly time-scale as the interval of interest. To do so, the data from the biweekly collections must be combined in some coherent fashion. The program DataTool combines periods by using the logical “OR”.

Begin DataTools. Go to the Period Frame (Other Frames -> Period Frame). To create a new time-scale, one that combines the biweekly measures into monthly measures, change the buttons under Output Period. The first two Input Periods will correspond to the first Output Period, so change the buttons from “1” and “2” to “2” and “2”. Similarly, the last two Input Periods will correspond to the second Output Period.

(A)	(B)
<pre> Data Simulated telemetry data from telem_sim_win npop 6 nper 4 gcov name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 gcov name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000 gcov name min time 1 19 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 gcov name distance time 0 116 176 235 345 467 748 icov indcov1 icov indcov2 ntaa 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 idlen 0 data 1 1 1 0 0 32.023027 4.214295 1 1 1 1 0 43.662921 4.081977 1 1 0 0 0 16.680651 5.554117 1 1 0 0 0 57.923423 4.608262 1 1 1 0 0 22.505242 3.736325 1 1 0 0 0 64.024929 6.446260 </pre>	<pre> Data Simulated telemetry data from telem_sim_win number-of-populations 6 number-of-periods 4 group-covariate name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 group-covariate name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000 group-covariate name min time 1 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 group-covariate name distance time 0 116.000000 176.000000 235.000000 345.000000 467.000000 748.000000 individual-covariate indcov1 individual-covariate indcov2 individual-covariate indcov.12 number-taaad 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 length_of_ids 0 full_hist data # population 1 1 1 1 0 0 32.023 4.21429 134.954 1 1 1 1 0 43.6629 4.08198 178.231 </pre>

Figure 9.2 Comparison of the (A) original **datafile** (i.e., **dtool.dat**) to the (B) transformed **datafile** (i.e., **dtool.c.dat**).

Change the buttons from “3” and “4” to “4” and “4”. Leave all the checkboxes checked. Save the new datafile under “dtool.d.dat”.



The new datafile (i.e., “dtool.d.dat”) has been changed in three places due to the transformation (Figure 9.3). The most obvious change is the reduction in periods. As stated before, the transformation uses the logical “OR” to combine periods (e.g., “1” OR “1” = “1”, “0” or “1” = “1”, “1” OR “0” = “1”, “0” or “0” = “0”). Thus, any individual that was released in either the first or second periods would get a “1”, and similarly, any individual that was recaptured in either the third or fourth periods would get a “1”. The second place the datafile has been changed is in the “number-tagged”. Notice that since the first and second release periods were combined, the number tagged was summed (e.g., $50 + 30 = 80$), and recorded as released in the first period. The third place the datafile has been changed resulted in some unforeseen consequences. Notice that the time-varying group-covariate “min” has been changed. Initially (Figure 9.3 (A)), this variable was the minimum temperature during the biweekly interval. However, after the transformation (Figure 9.3(B)), DataTool has replaced the minimum temperature with the average minimum temperature for the monthly time-scale. This transformation is probably not desired, and the user must go in and change this before further analyses are conducted.

9.3.4 Population Transforms

Suppose that the user originally collected data on 6 populations, (e.g., 3 Blocks x 2 Treatments) but after the original analysis, (s)he decided that all three blocks behaved similarly.

(A)

(B)

(A)	(B)
Data Simulated telemetry data from telem_sim_win	Data Simulated telemetry data from telem_sim_win
npop 6 nper 4	number_of_populations 6 number-of-periods 2
gpcov name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000	group_covariate name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
gpcov name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000	group_covariate name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000
gpcov name min time 1 18 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 13 4 13 -7	group_covariate name min time 1 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000
gpcov name distance time 0 116 176 235 345 467 748	group_covariate name distance time 0 116.000000 176.000000 235.000000 345.000000 467.000000 748.000000
icov indcov1 icov indcov2	individual-covariate indcov1 individual-covariate indcov2
ntag 50 30 0 0 50 30 0 0	number_tagged 80 0 80 0 80 0 80 0 80 0 80 0 80 0 80 0
idlen 0 data 1 1 1 0 0 32.023027 4.214295 1 1 1 1 0 43.662921 4.081977 1 1 0 0 0 18.680851 5.554117 1 1 0 0 0 57.923423 4.608262 1 1 1 0 0 22.505242 3.738325 1 1 0 0 0 64.024929 6.446260 1 1 1 1 0 62.999350 5.897147 1 1 0 0 0 60.674568 5.107279 1 1 1 1 0 60.303010 4.443959	length_of_ids 0 full_hist data # population 1 1 1 0 32.023 4.21429 1 1 0 43.6628 4.08198 1 0 0 18.6809 5.55412 1 0 0 57.9234 4.60826 1 1 0 22.5052 3.73833 1 0 0 64.0243 6.44626

Figure 9.3 Comparison of the (A) original **datafile** (i.e., **dtool.dat**) to the (B) transformed **datafile** (i.e., **dtool.d.dat**).

Thus, to increase the power of the calculations, the user may wish to **collapse** the current **6-population** design into a **2-population** design. Note, however, that subsequent analyses ignore any site-to-site differences in survival (i.e., σ_S^2) probabilities, and treatment comparisons are based on measurement error alone. To collapse the current 6-population design into a 2-population design, the user could use the Population Transformation Frame (Other Frames - > Population Frame). Assume that the variable named “gpcov1” indicates the Treatments and the variable “gpcov2” indicates the Blocks.

Begin DataTools. Go to the Population Frame (Other Frames -> Population Frame). To create the new population designations, one that combines the three blocks into a single block, change the buttons under Output Period. The first three Input Populations will correspond to the first Output Population, so change the buttons from “1” , “2” and “3” to “3” , “3” and “3” . Similarly, the last three Input Populations will correspond to the second Output Population. Change the buttons from “4” , “5” and “6” to “6” , “6” and “6” . Leave all the checkboxes checked. Save the new datafile under “dtool.e.dat” .

Input Population	Output Population	Include Group Covariates
1	3	<input checked="" type="checkbox"/>
2	3	<input checked="" type="checkbox"/>
3	3	<input checked="" type="checkbox"/>
4	6	<input checked="" type="checkbox"/>
5	6	<input checked="" type="checkbox"/>
6	6	<input checked="" type="checkbox"/>

The new datafile (i.e., “dtool.e.dat”) has been changed in several places due to the transformation (Figure 9.4). The most obvious change is the reduction in populations (i.e., from 6 to 2). As stated before, the transformation uses the logical “OR” to combine populations. Thus, any individual that was released as part of either the first, second or third original populations would be combined to form the first population in the new data, and similarly, any individual that was released as part of either the fourth, fifth or sixth original populations would be combined to form the second population in the new data. A second place the datafile has been

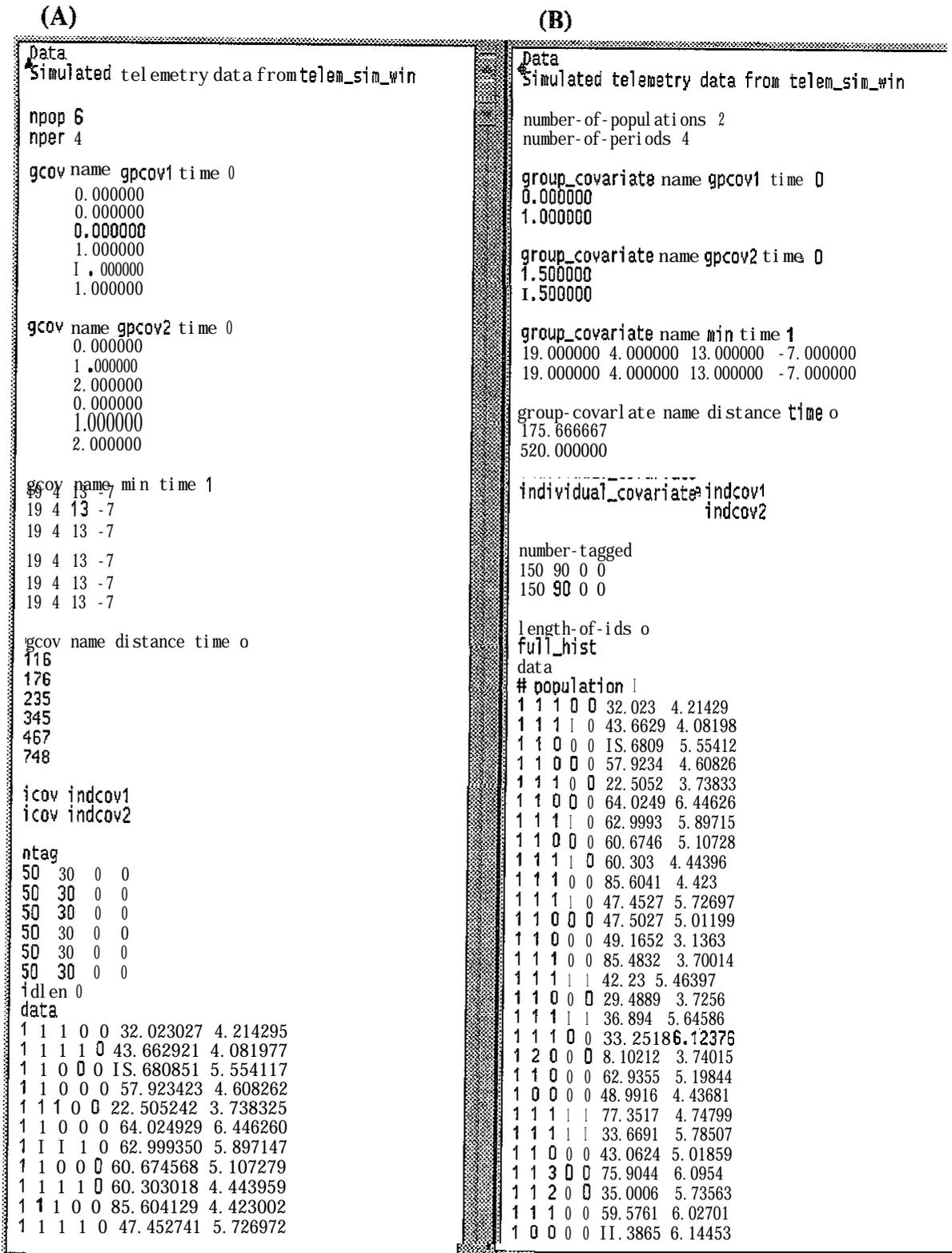


Figure 9.4 Comparison of the (A) original datafile (i.e., dtool.dat) to the (B) transformed datafile (i.e., dtool.e.dat).

changed is in the “number-tagged”. Notice that since the first three populations were combined, the number tagged was summed (e.g., $50 + 50 + 50 = 150$ for the first period, and 90 for the second period), and recorded as released as the first population.

Notice that because the group-covariate checkboxes were checked, all of the group covariates have been averaged across the populations. Thus, for the first group covariate, the first three original populations were from 1 treatment group, and all had a “0”. The average of the three “0’s” is a zero, so this is desired. Similarly, the second group covariate was the covariate we over which we combined populations (e.g., Blocks). We had already assumed that there was no effect of the second covariate (e.g., Blocks), and the identical values of 1.5 indicate that the populations have identical effects of “**gpcov2**”. The effect of the transformation on the group covariate “min” was to average the value across populations, and since all populations shared the same values per period, the only effect was to reduce the number of lines from “6” to “2”. Finally, the group covariate “distance” has been averaged. The user must decide whether this variable is now useful.

These examples have illustrated the use of DataTool. Although each example only used a single transformation, it is possible to combine periods, populations, group and individual covariates concurrently. However, as noted in the Period and Population Transformation Frames, sometimes the results from these transformations are not exactly what was desired. Therefore, the user may wish to limit the transformation to a single frame at a time. This will allow the user to verify more easily that the output datafile is desirable and sensible.

9.4 Conclusion

DataTool provides a coherent framework for transforming data from SURPH-readable files into new SURPH-readable files. DataTool transforms data at the individual and group-covariate level, and in addition, allows the user to combine data across periods and/or populations. This powerful tool should facilitate manipulations of datafiles used in **SURPH.1**.

Chapter 10

SURPH-PC Program

10.1 Introduction

Surph-PC is the version of SURPH.1 that runs under the MS-Windows@ operating system. All estimation and hypothesis-testing algorithms are identical to those used in the UNIX version. This chapter will familiarize the user with cosmetic and procedural differences present within the MS-Windows@ operating system. This chapter does not replace Chapter 7 of the SURPH.1 manual. Instead, this chapter is intended to clarify Chapter 7 with respect to differences in Surph-PC. The page numbers that follow each new topic indicate where more detailed information may be obtained. The order of the chapter will, where possible, follow the order dictated by the buttons on the Surph-PC Base Window. For a full coverage of SURPH.1 (i.e., statistical properties, potential uses, examples, and functionality) please read Chapters 1-9 of the SURPH.1 manual.

10.2 Differences in Surph-PC

Base Window (pp. 7.27-7.30) -The Surph-PC Base Window (Figure 10.1a) is cosmetically different from the UNIX version (Figure 10.1b). To begin with, some buttons are labelled differently in the two versions. The Surph-PC labels and their UNIX analogues can be found in Table 10.1. Additionally, the upper portion of the Base Window looks quite different. In the UNIX version, the ID, Name, LnLik, # Parameters, S Link, P Link and AIC are displayed for both the model selected in the Master Model List, and the model selected in the Testable Model List. In Surph-PC, this information is displayed only for the model selected in the Master Model List.

Likelihood Ratio Tests (pp. 7.21-7.22) - In Surph-PC, the results of LRT are displayed in a small box on the lower right-hand portion of the Surph-PC Base Window.

Load Data, Load Models, Store Models (pp. 7.33-7.38) - As in the UNIX version of SURPH.1, these functions reside on the File PulldownMenu. However, whereas in the UNIX version the user is prompted to type the directory and filename in fields on a pop-up window, in

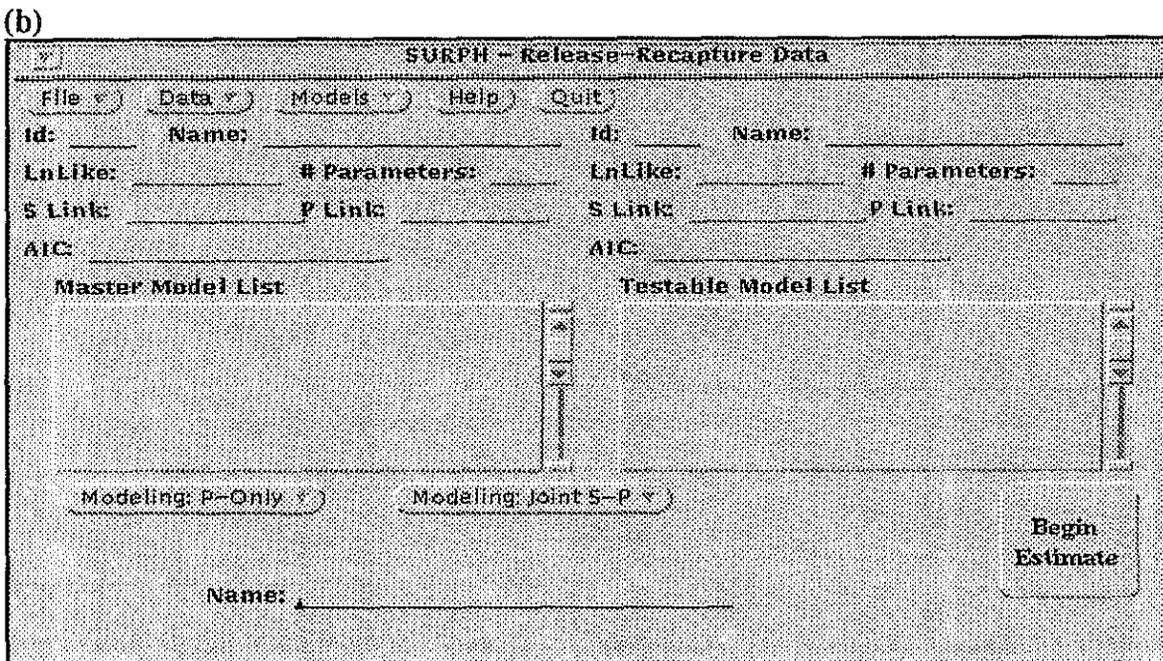
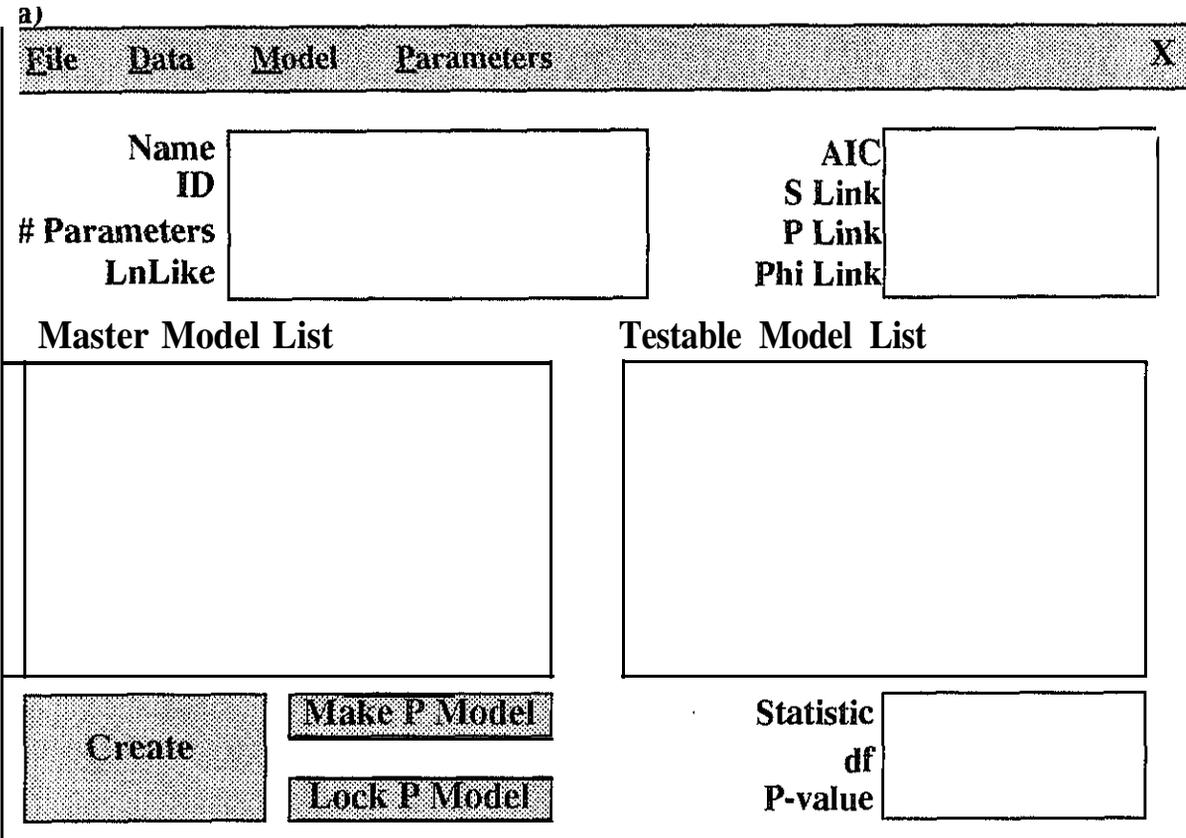


Figure 10.1 Surph Base Window for the (a) PC-Windows and the (b) Unix operating systems.

Table 10.1 Base-Window Button Labels. The button label from the UNIX-version of SURPH is given in the left-hand column, along with its PC analogue in the right-hand column.

UNIX	Surph-PC
File	File
Data	Data
Models	Parameters
Help	(Not Available)
Quit	Left-click on “Quit Window” icon (X) in upper right corner.
Modeling: P-Only	Parameters
Lock	Lock P Model
Modeling: Joint S-P	Parameters
Begin Estimate	Create

Surph-PC, these operations are preformed through point-and-click mouse operations. During “load” and “store” operations, Surph-PC expects data files to use the “.dat” suffix, and model files to use the “.mod” suffix.

ANODEV (pp 7.111 - 7.122) - The ANODEV option resides on the File Pulldown Menu in Surph-PC. As with the Surph-PC Base Window, some buttons on the ANODEV Window have different labels in the Surph-PC version. The Surph-PC labels and their UNIX analogues can be found in Table 10.2.

There are cosmetic changes to ANODEV as well. In Surph-PC, the ANODEV Window (Figure 10.2) has been split into two separate pop-up windows. The main window (Figure 10.2a) contains the information about the models that have been fit, the models that are used to compute the ANODEV Table, and shows the computed ANODEV Table. The second window (Figure 10.2b) is used to create and name additional models. In addition, quick-buttons have been added

Table 10.2 ANODEV Button Labels. The button label from the UNIX-version of SURPH is given in the left-hand column, along with its PC analogue in the right-hand column.

UNIX	Surph-PC
Calculate	Compute Table
Print	Print Table
Remove Model	Left-click on Transfer Bar
Transfer Model to Master List	Transfer Model
Quit	Left-click on "Quit Window" icon (X) in upper right corner.
Begin Estimation	Make Model

to the second window. In the UNIX version of ANODEV (Figure 10.3), these windows are combined.

A procedural difference between Surph-PC and the UNIX version is how models are moved from the Available Models region to the Testable Models region. In Surph-PC, to move a model from one region to the other, select the model (left-click), then left-click on the transfer bars (i.e., ).

Individual Covariates (pp. 7.47-7.56) - As in the UNIX version of SURPH.1, this function resides on the Data Pulldown Menu. A histogram of the data or a cumulative distribution plot of the data may be displayed using this selection.

The windows that are displayed for the PC version have some minor cosmetic differences compared to their UNIX analogues (Figures 10.4 - 10.7). First, the positioning of the various components displayed has been changed. However, all options that are available in the UNIX version are available in Surph-PC. Second, the slide bar that was used to change the "Occasion" designation in the UNIX version has been replaced by a pulldown menu in Surph-PC. Finally, the

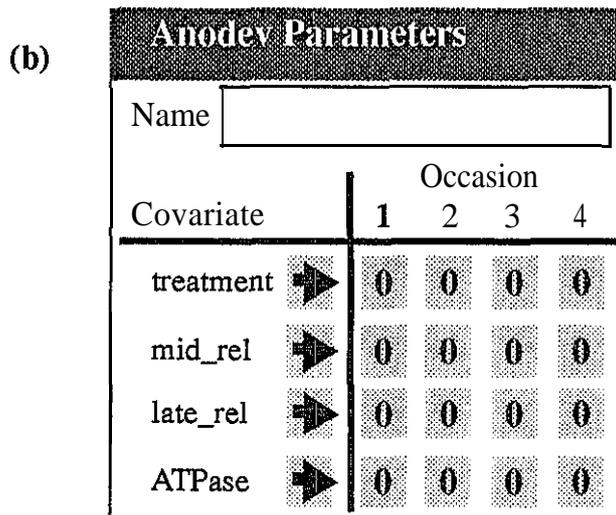
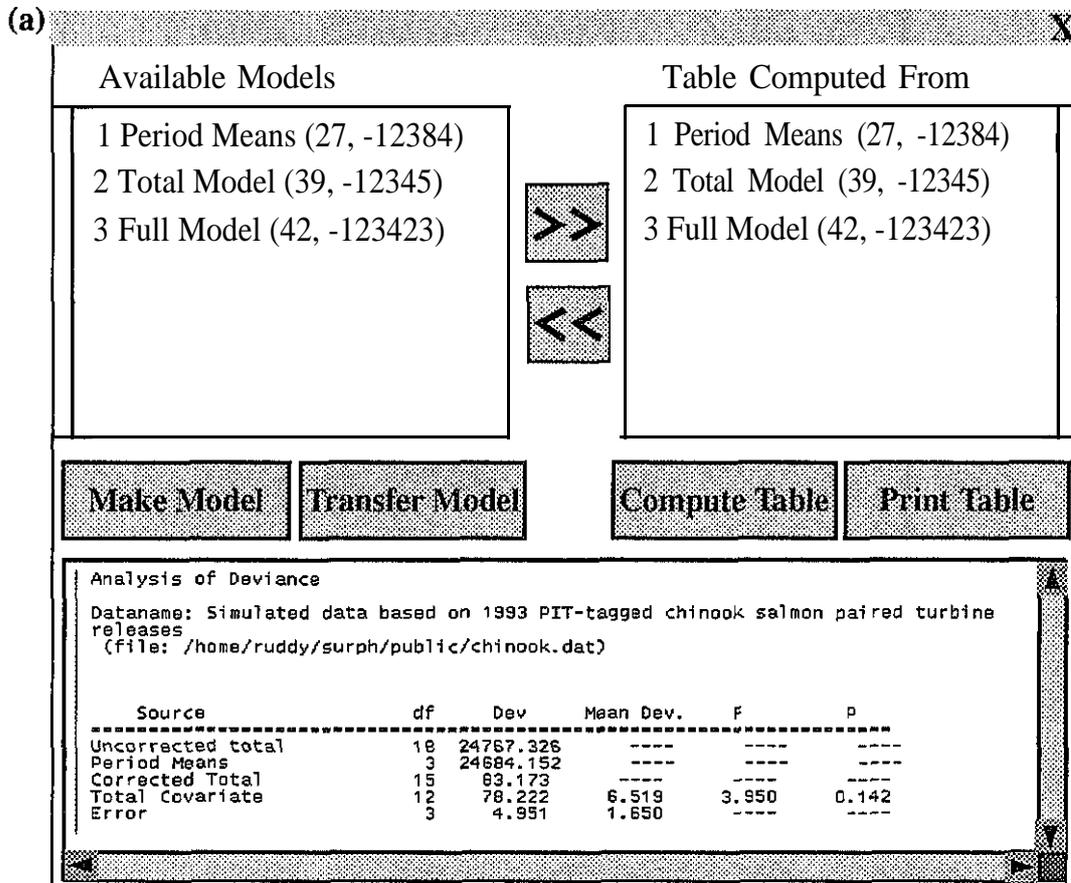


Figure 10.2 Surph-PC ANODEV Windows. The top figure (a) illustrates the ANODEV Table. Model names are displayed in the upper portion, whereas the ANODEV Table is displayed in the lower portion. The lower figure (b) is an ANODEV button pad. The ANODEV button pad is used to parameterize the group-covariate models.

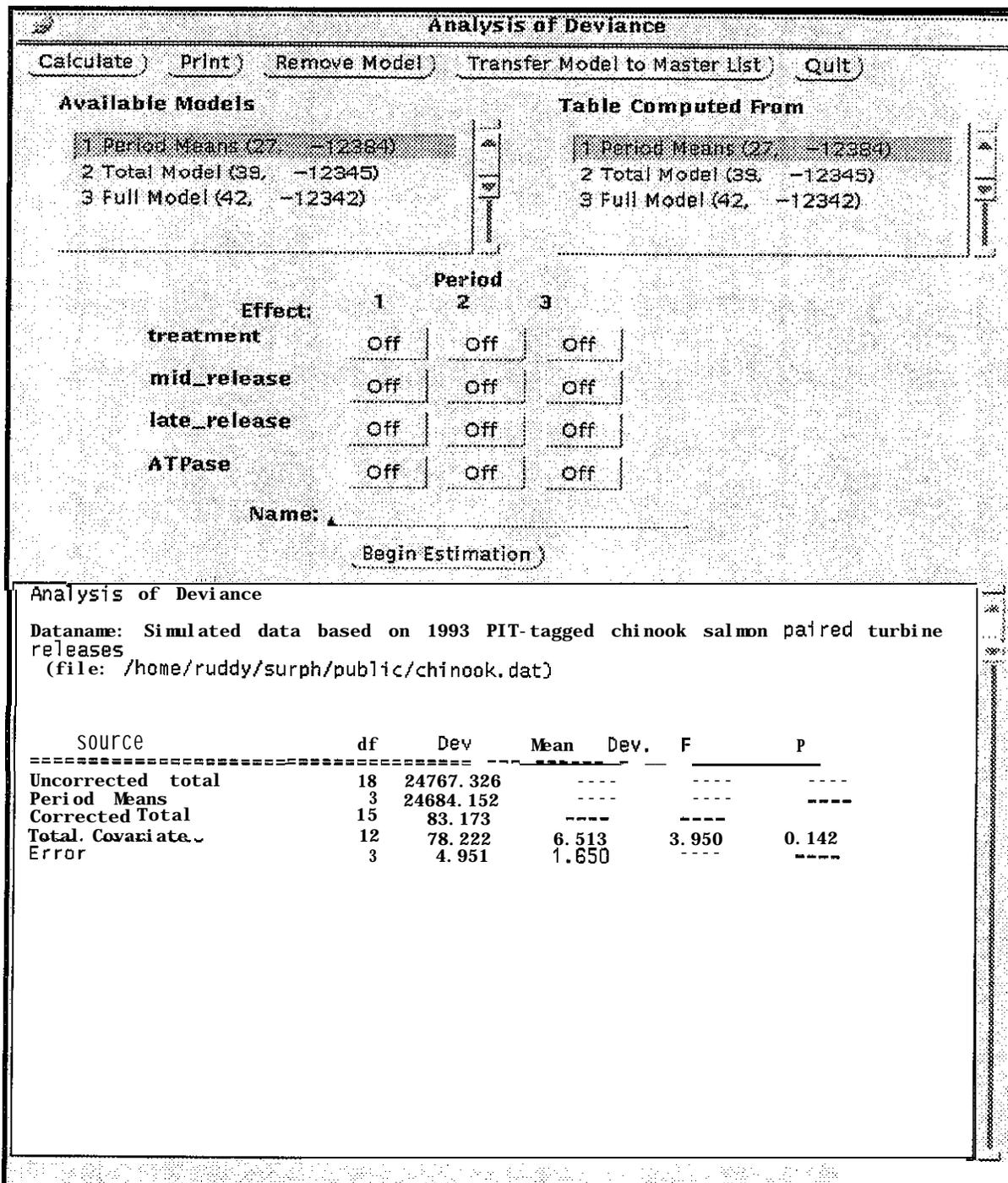


Figure 10.3 ANODEV Window from the Unix operating system. The major differences between the UNIX version and the PC version are that for the UNIX version (a) the ANODEV Button Pad is located within the ANODEV Window and (h) models are transferred differently.

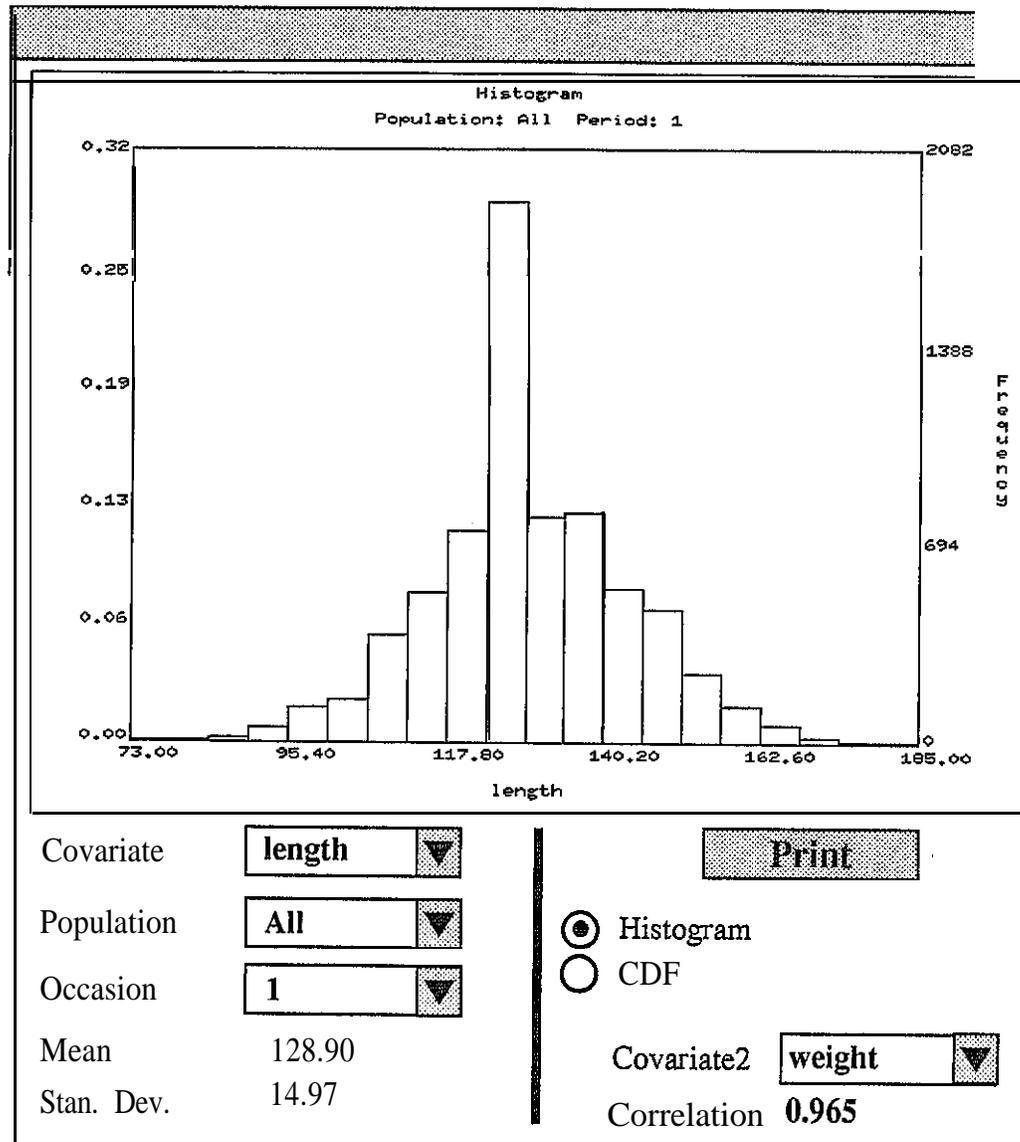


Figure 10.4 Individual Covariates Window for Snrph-PC - Histogram. A frequency histogram of the currently-selected covariate is displayed. On the left-hand side of the graphic, the relative frequency (probability) is given. On the right-hand side of the graphic, the frequency (count) is given. The horizontal axis denotes the value of the covariate.

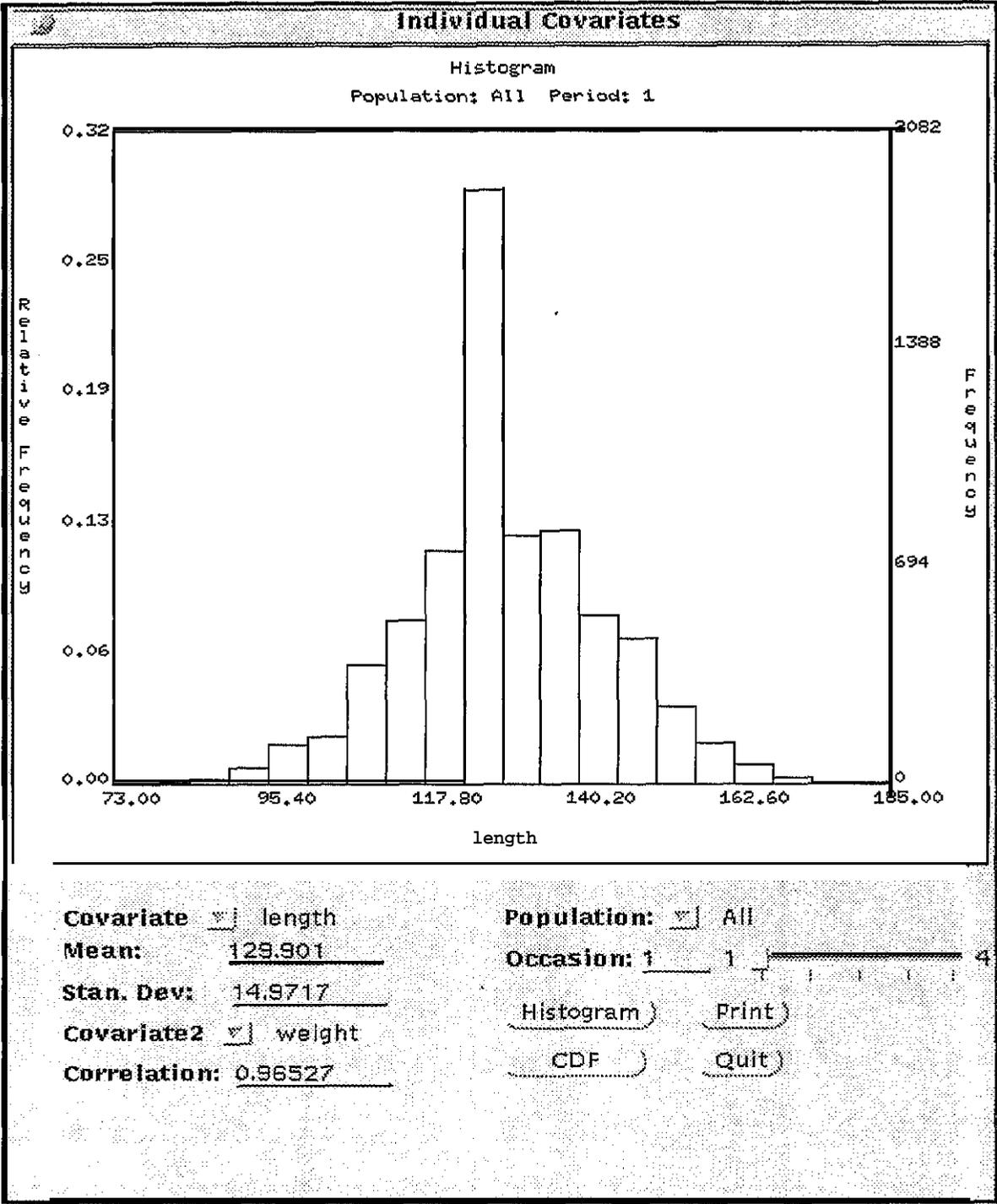


Figure 10.5 Individual Covariates Window for UNIX SURPH- Histogram. A frequency histogram of the currently-selected covariate is displayed. On the left-hand side of the graphic, the relative frequency (probability) is given. On the right-hand side of the graphic, the frequency (count) is given. The horizontal axis denotes the value of the covariate.

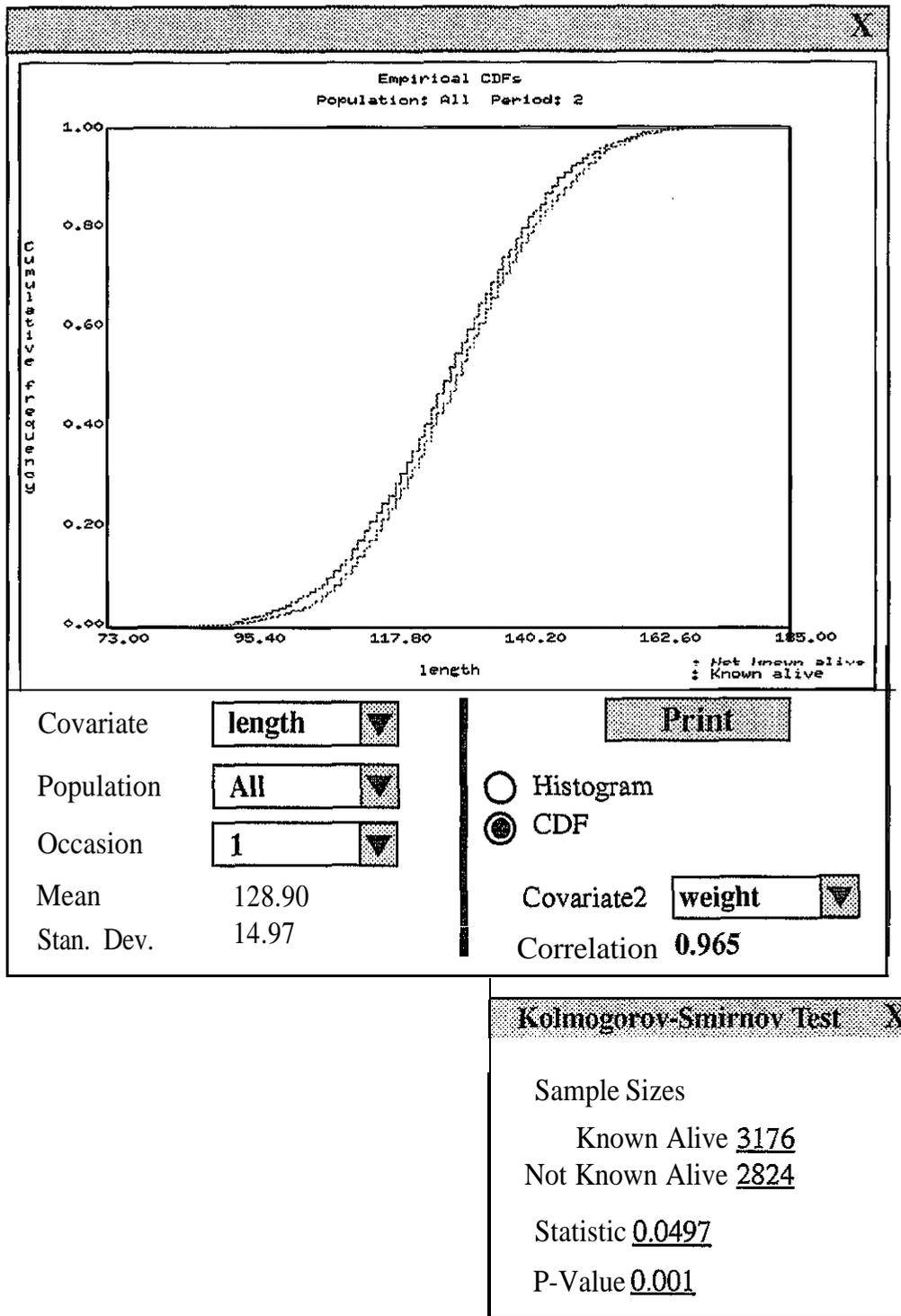


Figure 10.6 Individual Covariates Window for Snrph-PC. Cumulative Distribution Plot and K-S test window are displayed. The Cumulative Distribution Plots for those animals “Known Alive” and those animals “Not Know Alive” are displayed, along with the results of a K-S test.

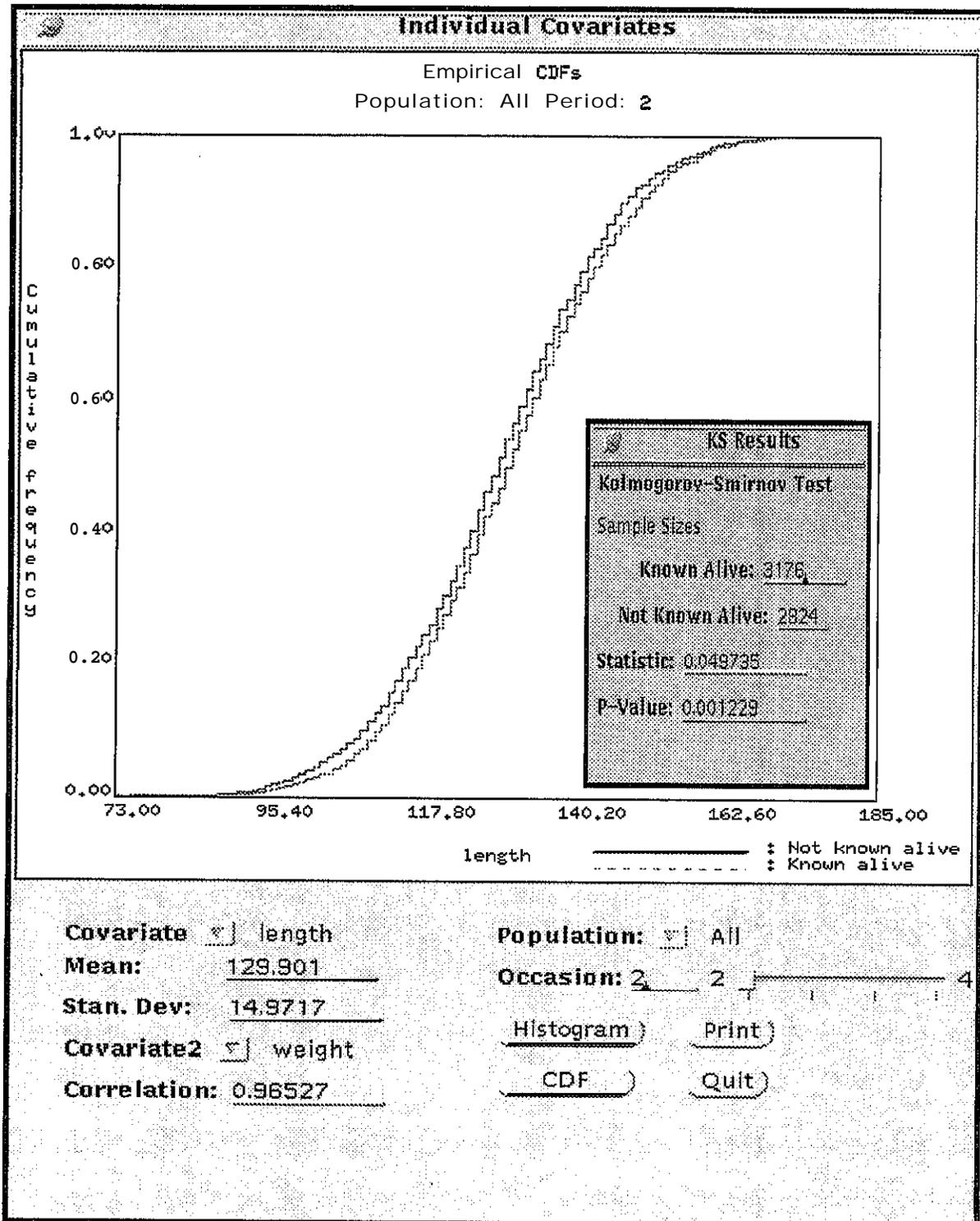


Figure 10.7 Individual Covariates Window for UNIX SURPH - Cumulative Distribution Plot and K-S test window. The Cumulative Distribution Plots for those animals “Known Alive” and those animals “Not Know Alive” are displayed, along with the results of a K-S test.

Quit Button has been removed. In Surph-PC the user must left-click on the “Quit Window” icon (i.e., X) in the upper right-hand corner to quit.

Capture Modeling (pp. 7.59-7.90) - One obvious change in Surph-PC is that all of the Button Pads are accessed using the Parameters Pulldown Menu. Thus, in Surph-PC, the difference between P-Only Modeling and Joint S-P Modeling is defined by which button is depressed prior to estimation of the model (i.e., Create vs. Make P Model), not which Button Pads are accessed. To initiate P-Only Modeling in Surph-PC, the user accesses the Capture Button Pads from the Parameters Pulldown Menu. The user then selects the Make P Model button on the Surph-PC Base Window to estimate the model. In UNIX, to initiate P-Only modeling, the user would select the Capture Button Pads from the Modeling: P-Only Pulldown Menu.

Renaming Models (pp. 7.173-7.174) - In the UNIX version, a Name field is provided on the SLJRPH Base Window prior to estimation. In Surph-PC, the name is initially fixed. The name can only be changed after estimation using the Rename command (Model -> Rename).

Survival, Capture and Product Curves (pp. 7.139-7.168) - As in the UNIX version of SURPH.1, these functions reside on the Model Pulldown Menu. The graphics that are displayed (i.e., group and individual curves) have some minor cosmetic differences. Only the initial pop-up window is displayed for each covariate type (i.e., group or individual) as all the cosmetic changes have occurred to the pop-up window itself, not the graphic within the pop-up window.

For the window that displays the group-covariate curves (Figures 10.8a and 10.8b), the buttons have been moved. In Surph-PC, the buttons are located on the upper portion of the pop-up window, whereas in the Unix version of SURPH, the buttons are located on the lower portion of the pop-up window. Otherwise the pop-up window is almost identical.

For the window that displays the individual-covariate curves (Figures 10.9 and 10.10), there are only cosmetic differences. Primarily, the location of the buttons that select between the various SURPH-curve options has been altered. In Surph-PC, the buttons are located on the left-hand side of the window, and label is adjacent to the button. On the UNIX version, the label of the graphic to be displayed is atop the button. All of the buttons except for the Quit Button are

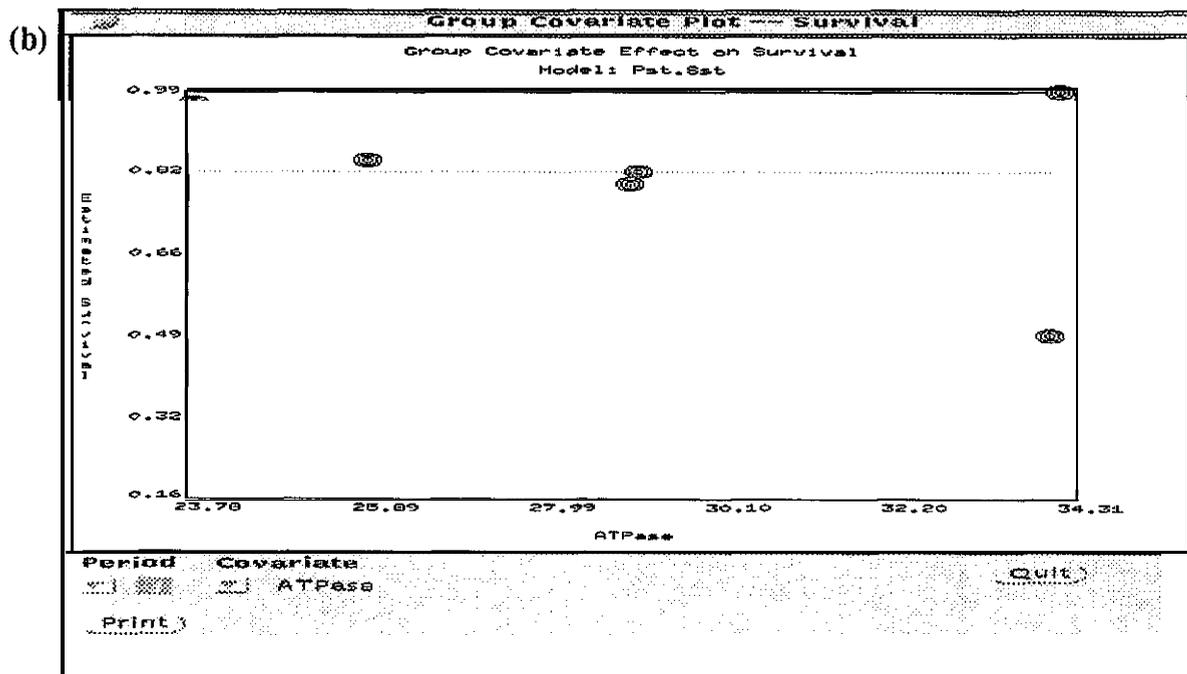
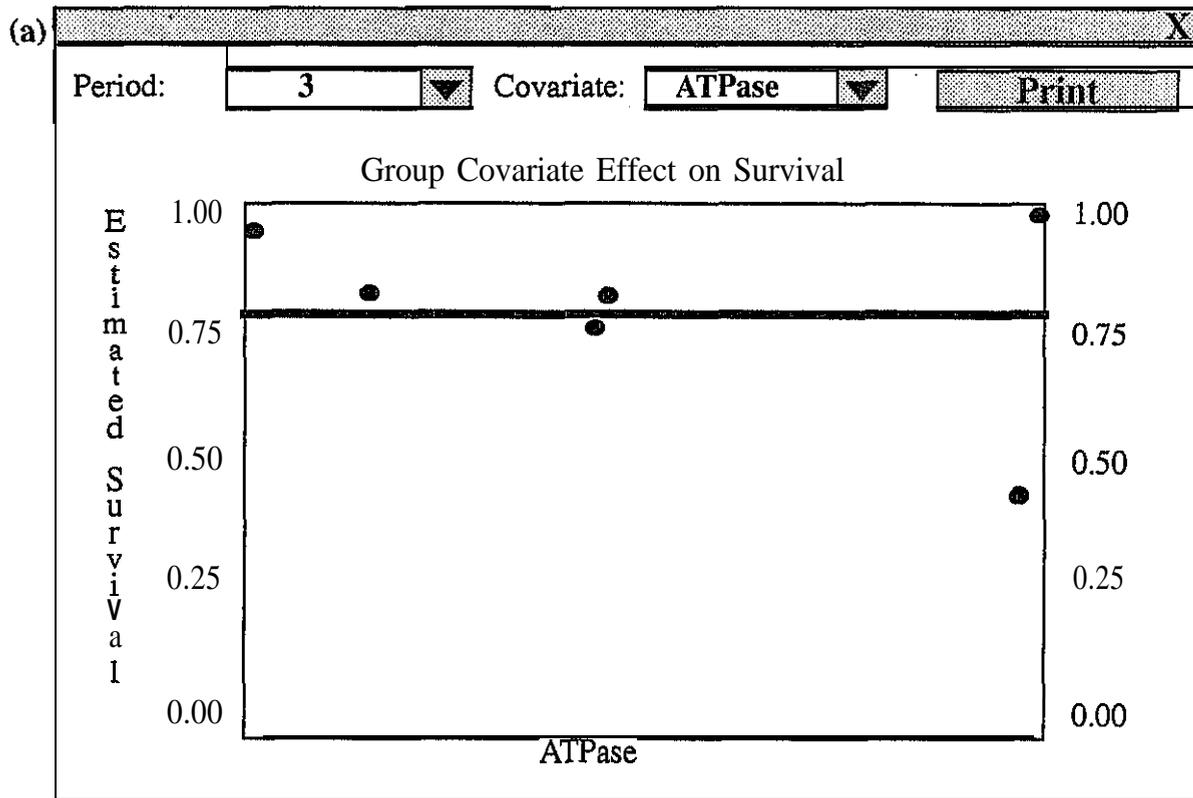


Figure 10.8 Model Graphics - Group Curves for (a) Surph-PC and for (h) the UNIX version. The vertical axis is the estimated survival. The horizontal axis spans the values of the group covariate. The fitted SURPH-model is displayed as a smooth curve. The CJS estimate of survival for each population is displayed as a point.

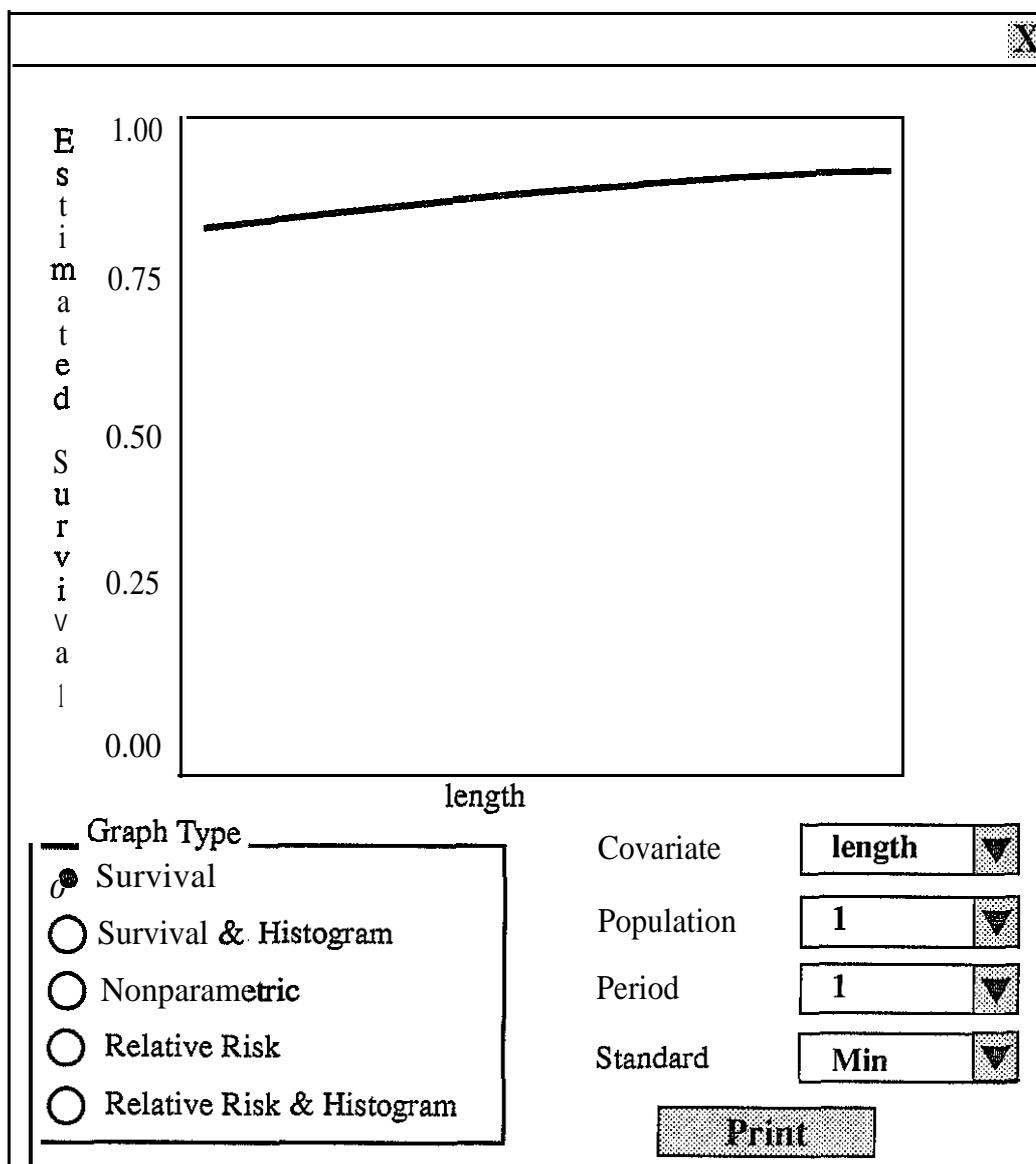


Figure 10.9 Models Graphics • Individual SURPH Curves for Surph-PC. The vertical axis is the estimated survival. The horizontal axis spans the value of the covariate (i.e., length). This example graphic illustrates the curve obtained from the fitted SURPH-model. The user may also view the survival curve with the frequency histogram or with the nonparametric estimates of survival. Alternatively, the user may plot the relative risk curve, with or without the frequency histogram.

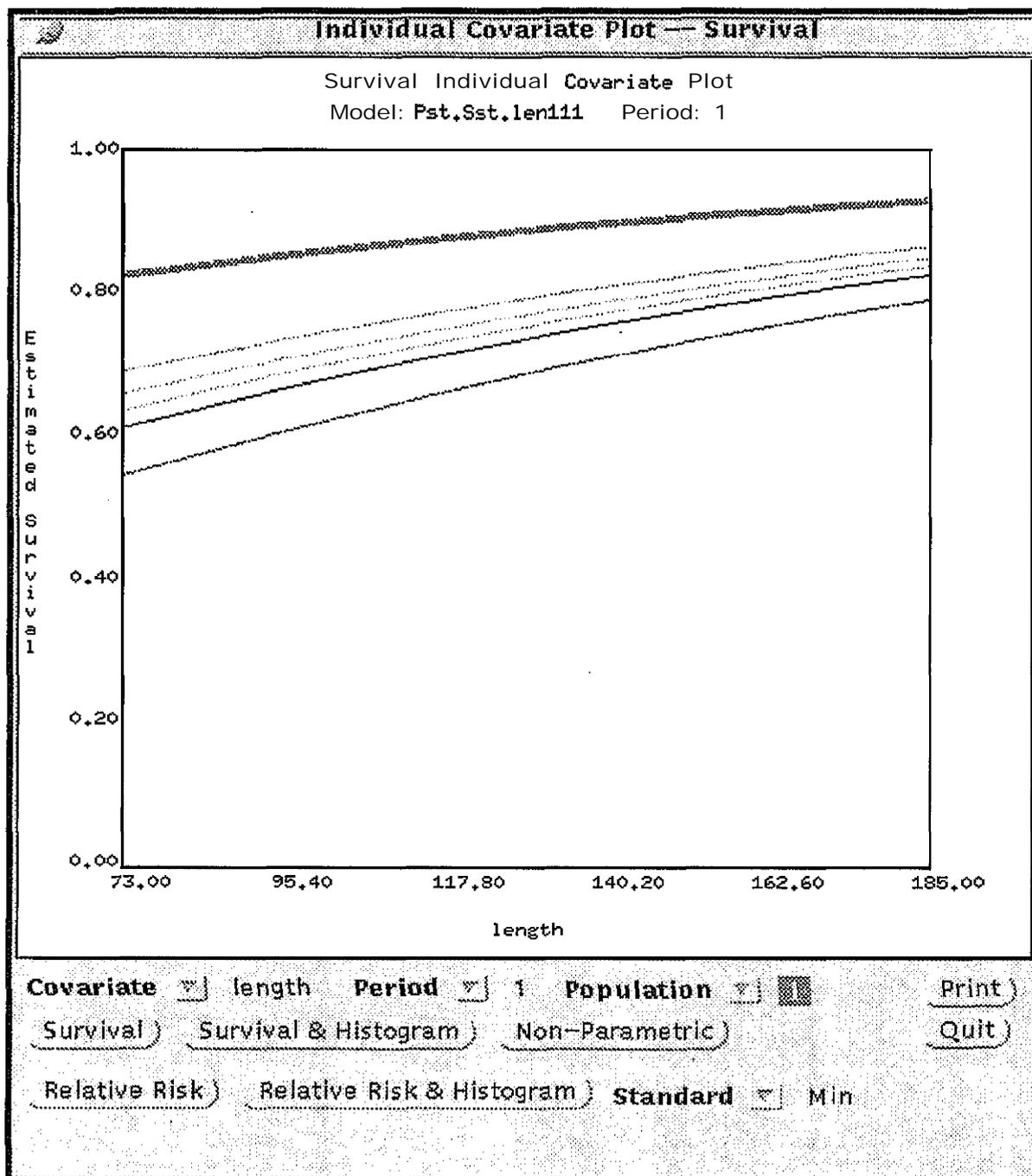


Figure 10.10 Models Graphics • Individual SURPH Curves for the UNIX version of SURPH. The vertical axis is the estimated survival. The horizontal axis spans the value of the covariate (i.e., length). This example graphic illustrates the curve obtained from the fitted SURPH-model. The user may also view the survival curve with the frequency histogram or with the nonparametric estimates of survival. Alternatively, the user may plot the relative risk curve, with or without the frequency histogram.

present. Similarly, the pulldown menus have been relocated. In Surph-PC, all pulldowns are located on the right-hand side of the window, whereas in the UNIX version, the pulldown menus were located above the graphics buttons.

Any other questions concerning **Surph-PC** should be clarified by reading Chapter 7 in the SURPH.I manual.

≡ 700
w/ tabs.

UNIVERSITY OF WASHINGTON
Seattle WASHINGTON 98195

Columbia Basin Research

September 6, 1996

list 368

Ms. Nancy Spencer (EWP)
U.S. Department of Energy
Bonneville Power Administration
905 NE 11th Avenue
Portland, OR 97232

Dear Nancy:

Enclosed, as we discussed by phone, are Chapters 8-10--additional chapters for the SURPH.1 Manual. I have ordered 300 copies of the tab inserts, which I will have sent directly to you for distribution with the chapter copies.

If you have any questions, please feel free to contact me by phone or email.

Sincerely,



Cindy Helfrich
Administrative Assistant

Enclosures

**Columbia Basin Research. School of Fisheries, University of Washington
Puget Sound Plaza Building, 1325 Fourth Avenue, Seattle, WA 98101-2509
Telephone (206) 685-1995. Fax (206) 616-7452**

UNIVERSITY OF WASHINGTON
SEATTLE, WASHINGTON 98195

Columbia Basin Research

December 17, 1996

Ms. Nancy Spencer
Bonneville Power Administration - EWP
P.O. Box 3621
Portland, OR 97208

Dear Ms. Spencer:

Enclosed please find the dividers for the three additional chapters to the SURPH.1 Manual:

Data Tools Program
Sample Size Program
SURPH-PC

Please arrange for distribution with the chapters. Extra copies can be returned to me. Thanks in advance for your assistance.

Sincerely,



Cindy Helfrich
Administrative Assistant

Poe, Pat - EWI

From: cindy
To: Pat Poe
Subject: SURPH Chapters
Date: Wednesday, August 21, 1996 10:35AM

I have three additional chapters ready for release for SURPH:

- Chapter 8: Sample Size Subprogram
- Chapter 9: Data Management Subprogram
- Chapter 10: SURPH-PC

I did not coordinate the first distribution of SURPH, but my understanding is that the GPO will handle the contracting of the printing, since we will request more than 50 copies of these deliverables. Is this still correct? I believe the initial run of SURPH.1 was 300 copies, with 100 sent here. Is Nancy Spencer the person I should contact? (Would you please provide her e-mail address or phone number?)

I will also need to order additional binder tabs through our printing services, and then forward them to your printer for distribution.

I will also need to update Mailing List Code #368 prior to distribution.

Would you like copies of these chapters prior to printing? Do this initial game plan sound appropriate?

Thanks.

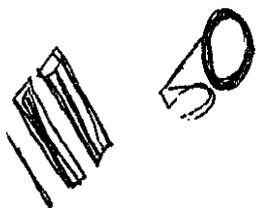
Cindy

Cross reference with # 112
Cindy had not sent changes to Distribution list yet as of 9/24.
PS

III [] III

315

1. 1





Chapter 8

Sample Size Program

8.1 Introduction

The success of a survival study can be measured, in part, by the level of certainty associated with the estimated survival probabilities. The greater the level of sampling precision, the greater the certainty the estimate reflects the true but unknown survival rate. Variance estimates (i.e., $\text{Var}(\hat{S}|\mathcal{S})$) and standard errors (i.e., $\sqrt{\text{Var}(\hat{S}|\mathcal{S})} = \hat{SE}(\hat{S}|\mathcal{S})$) both express the measurement errors associated with **estimating** the survival rate of a **specific** release group **in** time and space.

The measurement error associated with a specific survival estimate will be a function of (a) actual survival probabilities, (b) capture probabilities, (c) release sizes of marked animals, and (d) rate marked animals are knowingly removed from the investigation. Anticipated measurement error for a survival study can be projected by having preliminary “guesstimates” of these quantities. Furthermore, investigators can determine required release sizes and capture probabilities to achieve a pre-specified level of sampling precision. This chapter explains how a utility associated with SURPH.1 called `SAMPLE_SIZE` can be used to investigate sample sizes and precision of release-recapture and known-fate studies whose focus is to estimate period-specific survival probabilities.

Neither the survival probabilities nor the capture probabilities **will** be known prior to beginning the study. As such, the user must make rough guesses as to the values of these quantities in sample size calculations. One suggested way to improve on these guesses is to use a pilot study (Skalski 1993). Because the survival and capture probabilities are often beyond the control of the experimenter, the primary tool the experimenter has for changing the precision is to alter the release size. The Sample-Size program can be used to investigate the interrelationships between release size, survival probability, capture probability, and known removals on the precision of the survival estimates.

8.2 Statistical Definition of Precision

Common measures of precision include the standard error of a parameter estimate (i.e., $\hat{SE}(\theta) = \sqrt{\widehat{Var}(\hat{\theta}|\theta)}$) and the coefficient of variation (i.e., $cv = \sqrt{\widehat{Var}(\hat{\theta}|\theta)} / \hat{\theta}$). The standard error expresses precision in absolute terms while the coefficient of variation expresses precision in relative terms. A third approach expresses precision in a probabilistic manner reflecting the fact that the parameter estimate is a random variable. In estimating survival probabilities, a useful measure of precision is expressed as

$$P(|\hat{S} - S| < \epsilon) \approx 1 - \alpha \quad (8.1)$$

the probability that the absolute deviance between the estimate and true value (i.e., $|\hat{S} - S|$) is arbitrarily small (i.e., ϵ) with some high probability of occurrence (i.e., $1 - \alpha$). An alternative measure of precision,

$$P\left(\frac{|\hat{S} - S|}{S} < \epsilon\right) = 1 - \alpha \quad (8.2)$$

expresses the deviance in relative terms (i.e., $(\hat{S} - S) / S$). Expression (8.2), while a valid expression of precision, is less likely to be used in estimating survival because survival is already a relative measure from 0-1.

For example, if there was a desire for a study to estimate survival (S) **within ± 0.10 , 80** percent of the time, expression (8.1) would be rewritten as

$$P(|\hat{S} - S| < 0.10) = 0.80$$

where $\epsilon = 0.10$ and $\alpha = 0.20$. The demands on precision increase as ϵ and/or α become smaller. An **investigator** must choose the most appropriate level of precision for the goals of the study. **Often** the investigator may want to evaluate alternative levels of study effort and the associated changes in precision that may be anticipated.

precision (**i.e.**, expression 8.1) can also be interpreted where ϵ is the half-width of an anticipated $(1 - \alpha)$ 100% confidence interval estimate of \hat{S} . Values of ϵ and α can therefore be selected to achieve the desired width (ϵ) of a confidence interval at the significance level. The relationship between the precision expression (8.1) and a $(1 - \alpha)$ 100% confidence interval can be seen from the following manipulations where

$$P(|\hat{S} - S| < \epsilon) = 1 - \alpha$$

$$P(-\epsilon < \hat{S} - S < \epsilon) = 1 - \alpha$$

$$P\left(\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} < \frac{\hat{S} - S}{\sqrt{\text{Var}(\hat{S}|S)}} < \frac{\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = 1 - \alpha$$

and assuming S normally distributed

$$P\left(\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} < Z < \frac{\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = 1 - \alpha$$

$$1 - 2P\left(Z < \frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = 1 - \alpha$$

$$P\left(Z < \frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = \frac{\alpha}{2}$$

$$\Phi\left(\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}}\right) = \frac{\alpha}{2}$$

$$\frac{-\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} = Z_{\frac{\alpha}{2}}$$

$$\frac{\epsilon}{\sqrt{\text{Var}(\hat{S}|S)}} = Z_{1 - \frac{\alpha}{2}}$$

$$\epsilon = Z_{1 - \frac{\alpha}{2}} \sqrt{\text{Var}(\hat{S}|S)}$$

where Z is a standard normal deviant (i.e., $P(|Z| > z_{1-\frac{\alpha}{2}}) = \alpha$).

In the Sample-Size program, two measures of precision will be presented. The two measures are the anticipated standard error (i.e., $\sqrt{Var(\hat{S}|S)}$) and values of ϵ and $1 - \alpha$ from probabilistic expression (8.1).

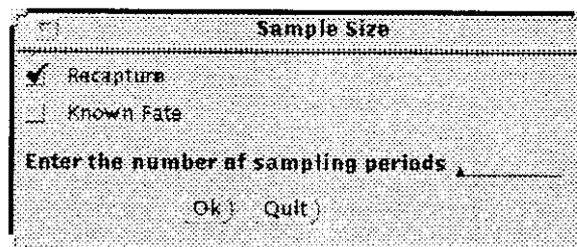
8.3 Using the Sample-Size Program in Designing a Single-Population Study

This section illustrates the **main** features of the Sample-Size program.

To begin the Sample Size program. at the prompt, type "sample":

> sample

Left-click on either the Recapture or the Known-Fate Check-Box. The Recapture Check-Box should be selected if the study that will be used is a release-recapture study. Select the Known-fate box if the study is a telemetry study. Enter the number of sampling occasions in the blank field. then select OK if the information is correct. The number of sampling occasions is equal to the number of times the user will attempt to detect marked animals. Thus, to calculate the number of sampling occasions, do not include the initial release. If something is incorrect. go back and change the selection before selecting OK. Select Quit to leave the Sample Size program



In a release-recapture study, both survival and capture probabilities are unknown random variables, so the user will be asked to provide "guesstimates" of both survival and capture probabilities during each period. In a known-fate study, only the survival probabilities are random variables (the probability of capture in a known-fate study is 1.00), therefore the user will be **asked** to provide estimates of only survival **probabilities** during each period.

After you have checked the appropriate study and entered the number of sampling periods, left-click on OK. The next window that will appear is the Sample Input Window. The Sample Input Window is tied to iteratively investigate the effects of changing sample size, survival and recapture probabilities.

Sample Input					
	S1:	S2:	S3:	S4:	S5:
R0:	R1:	R2:	R3:	R4:	
	P1:	P2:	P3:	P4:	P5:
	D1:	D2:	D3:	D4:	OK

The Sample Input Window is the heart of the Sample Size program. It is here that the user will alter the release sizes and possibly the capture probabilities to arrive at the level of effort needed to get the precision necessary for the study. Each input has a letter-number designation. The letter designates the type of value; the number designates the sampling occasion. Letter designations are **S** (for Survival Probability), **R** (for Release Number), **P** (for Capture Probability), and **D** (for Proportion Removed/Known Dead). The **R_j** field should be used to enter the number of unmarked animals that will be released on the **i**th occasion. The **S_j** field should be used to enter the estimated survival probability of the population between sampling occasions “j-1” and “j”. The **P_j** field should be used to enter the estimated capture probability of the population on sampling occasion “j”. And similarly, the **D_j** field should be used to enter the **estimated** proportion of the tagged animals that **are** captured that **will** be removed on sampling occasion “j”. This removal can be due to either accidental deaths associated with the capture process or due to purposeful removals.

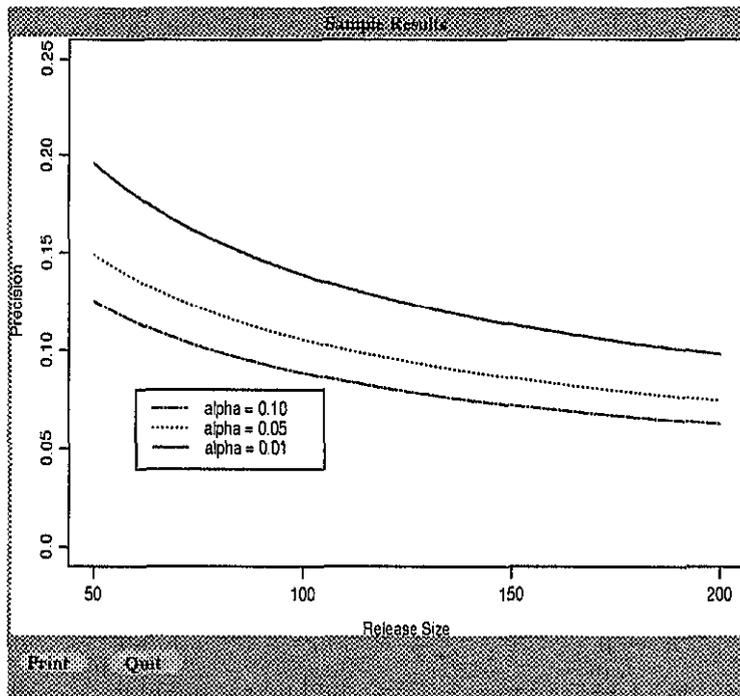
Fill in the blank fields with guesstimates of survival probabilities, capture probabilities, and number released. All fields must be filled before the user can proceed. Once all the blank fields have been filled, left-click on the OK.

The output windows are a table of estimated standard errors for the survival estimates and a graphical display relating precision (defined as half the width of a 1 - α confidence interval around the survival estimate) to the release size and the α -level. The estimates of the standard errors are computed using the closed form equations for the variance estimates based on the multinomial likelihood (Cormack 1964).

Two windows will appear, a text window and a graphics window. The text window displays the per-period survival, and the associated projected standard error.

Period	Survival Probability	Expected Std. Error
1	0.950	0.042
2	0.950	0.031
3	0.950	0.025
4	0.950	0.029

The graphics window displays the precision (defined as the half-width of the α -level confidence interval) as a function of the release size. There are three curves in the graphics window, associated with three α -levels (0.01, 0.05, and 0.10).



Sample size determination *now* becomes an iterative process. If the current release size(s) does not give sufficient precision, the user can either find a way to increase the capture probability(ies), or the user can increase the release size(s). Generally, in order to half the estimated standard error, the user must quadruple the release size. If there is low survival, the user may get better precision by going from a single-release to a multiple-release. This way, a

higher sample size is maintained throughout the study. If survival is high throughout the study, there is little to be gained by using multiple releases.

Example 1 - Release size to achieve a desired precision level

Suppose that in a particular survival study, the desired precision of the survival estimate is to be **within ± 0.05** of the true value 95% of the time ($\epsilon = 0.05, \alpha = 0.05$). The study will consist of a single release followed by 4 survey periods. The anticipated survival during any one period will be approximately 0.70, and the probability of capture will be about 0.40. No removals of tagged animals are planned. The question that can then be asked is how many animals must be released in order to achieve the desired level of precision.

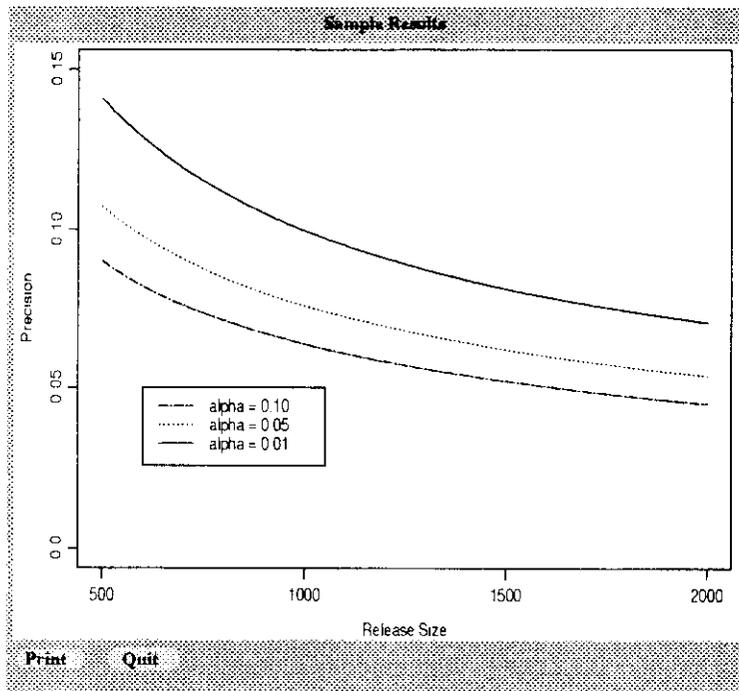
Begin the Sample_Size program. Check the Recapture check-box, and enter "4" in the Number of Sampling Periods field. Enter the approximate survival, capture and removal rates into the blank fields of the Sample Input Window. Once completed, the Sample Input Window should look like the window below.

Sample Input				
	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
RB: 1000	R1: 0	R2: 0	R3: 0	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	OK

Left-click on OK when all blanks have been filled.

Initially, we guess that 1000 animals maybe sufficient. However, when we examine the anticipated precision curve at a $\alpha = 0.05$, we discover that a release size of 1000 will yield a precision of about ± 0.08 (Figure 8.1 (A)). Using the "rule-of-thumb" discussed previously, the subsequent guesstimate of R_1 was quadrupled to a release size of 4000 (Figure 8.1 (B)), in an attempt to produce the desired level of precision in the first period. However, notice that although the precision is now within the desired range for the first period survival, the precision for both the second and third period survivals is outside the range desired (Figure 8.1 (C)). If we wish to maintain the precision desired throughout the study, we will either need to increase the initial release, or supplement the number of marked animals during subsequent releases.

(A)



(B)

Sample Input				
	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
R0: 4000	R1: 0	R2: 0	R3: 0	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	Ok

(C)

Sample Results		
Print	Quit	
Period	Survival Probability	Expected Std. Error
1	0.700	0.019
2	0.700	0.031
3	0.700	0.045

Figure 8.1 A selection of the windows from the first single-release example. (A) Graphical display of the initial guess (i.e., $R_0 = 1000$). (B) Sample Input Window for the intermediary guess (i.e., $R_0 = 4000$). (C) Textual output of the intermediate guess (i.e., $R_0 = 4,000$).

If we try to achieve the desired precision throughout the study by increasing the initial release, trial and error indicates that the initial release size must be near 14,000 animals. This release size drastically increases the precision for the first and second period survival estimates, but just achieves the precision desired during the third period (Figures 8.2 (A) and **(B)**).

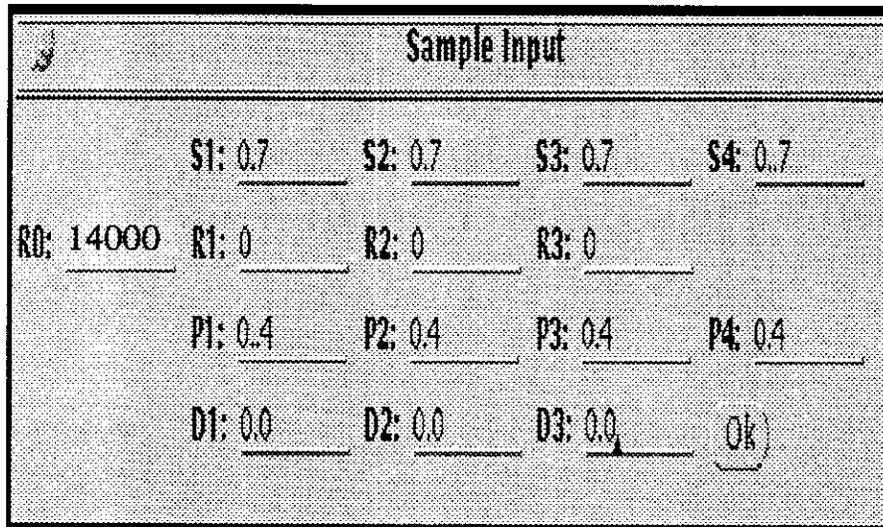
It may be useful to see what period-specific release sizes would be necessary to achieve the desired precision for each survival estimate. Using the information contained in Figure 8.1(A), use 2500 animals in the initial release. This gives the desired precision for the first period. Then begin adding animals to subsequent releases until the standard error for each of the per period survival estimates is below $\epsilon = 0.05$ (this is because at $\alpha = 0.05$, the half-width of the confidence interval is anticipated to be $1.96(\text{estimated standard error})$). Iterative trial and error indicates that releases of **1400, 1400, 1700** and 1700 would achieve the desired precision throughout the study (Figures 8.3 (A) and **(B)**). We conclude that a total of 6200 animals must be released under the multiple-release study design, whereas 14,000 animals were necessary under the single-release design. Thus, if the multiple-release study design is an option, less than half as many animals would have to be marked under the multiple-release study design compared to the single-release design to achieve the desired precision throughout the study.

Example 2 - Precision at a specified release size

In the previous example, the precision was pre-specified and the release sizes were determined to achieve the desired precision level. For many studies, however, the release sizes are fixed, either because of budgetary constraints or because of time and effort constraints. Therefore, a common issue is determining the level of precision likely to be achieved with a predetermined level of effort.

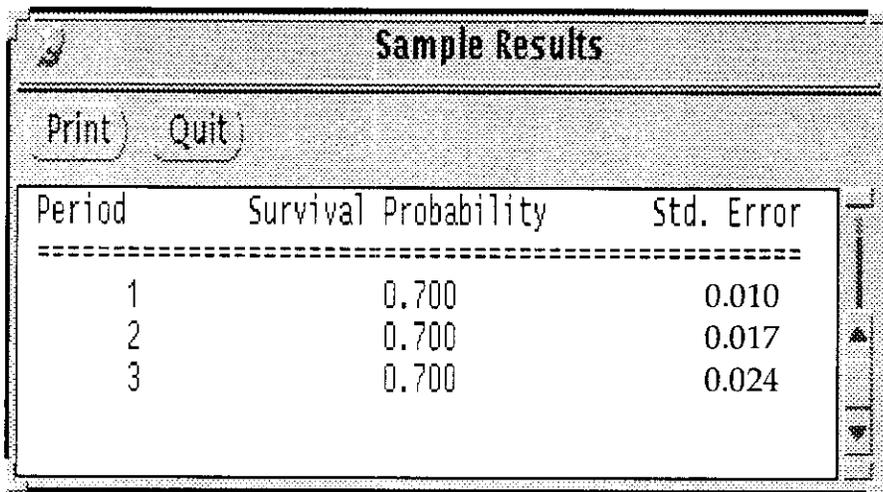
Assume that only an initial release of 14,000 marked animals is feasible. The study will be 4 periods long. The estimated survival during any one period will be approximately 0.70. and that the probability of capture will be about 0.40. No removals are anticipated. The question is, at the $\alpha = 0.05$ level, what is the anticipated level of precision (i.e., half-width of a 95% CI) that can be achieved with this level of effort.

(A)



The 'Sample Input' window displays various parameters for a single-release design. The parameters are arranged in a grid-like format with labels and values separated by colons. The values are: S1: 0.7, S2: 0.7, S3: 0.7, S4: 0.7; R0: 14000, R1: 0, R2: 0, R3: 0; P1: 0.4, P2: 0.4, P3: 0.4, P4: 0.4; D1: 0.0, D2: 0.0, D3: 0.0. There is an 'Ok' button in the bottom right corner.

(B)



The 'Sample Results' window shows a table of survival probabilities and standard errors for three periods. It includes 'Print' and 'Quit' buttons at the top left. The table data is as follows:

Period	Survival Probability	Std. Error
1	0.700	0.010
2	0.700	0.017
3	0.700	0.024

Figure 8.2 A selection of the windows from the first single-release example. (A) Graphical display of the Sample Input Window for the final guess (i.e., $R_0 = 14,000$) using a single-release design. (B) Textual output of the Sample Results Window for the final guess (i.e., $R_0 = 14,000$) using a single-release design.

(A)

Sample Input				
	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
R0: 1400	R1: 1400	R2: 1700	R3: 1700	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	Ok

(B)

Sample Results		
Print	Quit	
Period	Survival Probability	Std. Error
1	0.700	0.025
2	0.700	0.023
3	0.700	0.025

Figure 8.3 A selection of the windows from a multiple-release example. (A) **The** Sample Input Window for the Fixed Precision Example, with the final release sizes under a multiple-release design. (B) Textual output of the Sample Results Window for the Fixed Precision Example using a multiple-release design.

As before, fill in the blanks of the Sample Input Window with the known information (Figure 8.2 (A)). When finished, select OK.

Recall that at $\alpha = 0.05$, the half-width of the confidence interval will be approximately 1.96 times the estimated standard error. **Thus**, using this approximation, we find that the precision for the **first** period survival is anticipated to be $\epsilon = 0.022 (-1.96SE(\hat{S}))$, the precision for the, second period survival is anticipated to be $\epsilon = 0.033$. and the precision for the third period survival is anticipated to be $\epsilon = 0.049$ (Figure 8.2 (B)).

Assume **that** an alternative study design allows multiple releases of 1500 newly marked animals per period. Using the same survival and capture probabilities as before, at the $\alpha = 0.05$ level, what level of precision will be achieved?

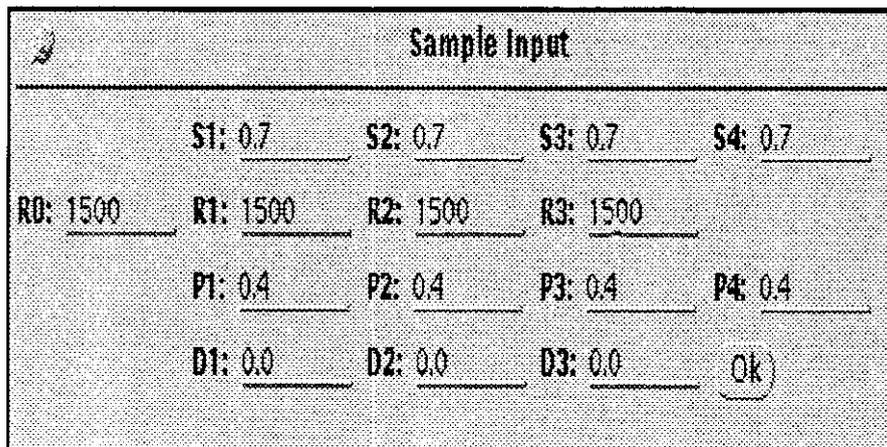
Fill in the blanks of the Input Window with the known information (Figure 8.4 (A)). When finished select OK.

Because we expect the parameter estimates to be distributed asymptotically-normal, the half-width of the confidence interval will be 1.96 times the estimated standard error. Using this fact, we find that the precision for the first period survival is anticipated to be $\epsilon = 0.047$, the precision for the second period survival is anticipated to be $\epsilon = 0.045$, and the precision for the first period survival is anticipated to be $\epsilon = 0.051$ (Figure 8.4 (B)).

8.4 Conclusion

The Sample Size program can be used to determine the level of effort needed to achieve a desired level of precision for \hat{S} or the precision that is attainable given a prescribed level of effort. As such, this tool should be used during the design phase of the survival study, **and** can be used in concert with the SURPH analysis of survival data.

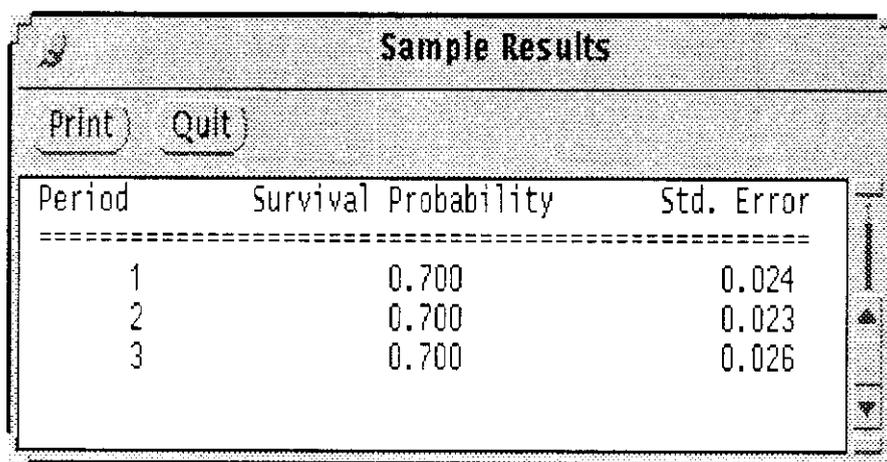
(A)



The 'Sample Input' window contains the following data:

Sample Input				
	S1: 0.7	S2: 0.7	S3: 0.7	S4: 0.7
RD: 1500	R1: 1500	R2: 1500	R3: 1500	
	P1: 0.4	P2: 0.4	P3: 0.4	P4: 0.4
	D1: 0.0	D2: 0.0	D3: 0.0	Ok

(B)



The 'Sample Results' window displays the following data:

Period	Survival Probability	Std. Error
1	0.700	0.024
2	0.700	0.023
3	0.700	0.026

Figure 8.4 A selection of the windows from the second multiple-release example. (A) The Sample Input Window for Fixed Effort Example. (B) Textual output of the Sample Results Window for the Fixed Effort Example using a multiple-release design.

The examples in this chapter dealt only with estimating the precision of survival estimates from the study of a single marked cohort. In many studies, the desire is to determine how survival rates differ between populations or are affected by environmental or human influences. Readers are encouraged to read Skalski and Robson (1992) and Burnham et al. (1987) before designing a multi-population study.



Chapter 9

Data Tools Program

9.1 Introduction

The original release of SURPH.1 (Release date: December 1994) incorporated the ability to transform data (Dat-> Data Transformations) within SURPH-readable datafiles. However, the Data Transformations utility had limited capabilities. Thus, we have created a more flexible and powerful database manager called DataTool. DataTool takes a SURPH-readable datafile, and creates a new SURPH-readable datafile that incorporates data manipulations of group and/or individual covariates. DataTool also allows the user to combine periods and populations within a datafile. This powerful tool should facilitate manipulations of the datafiles used in SURPH.1.

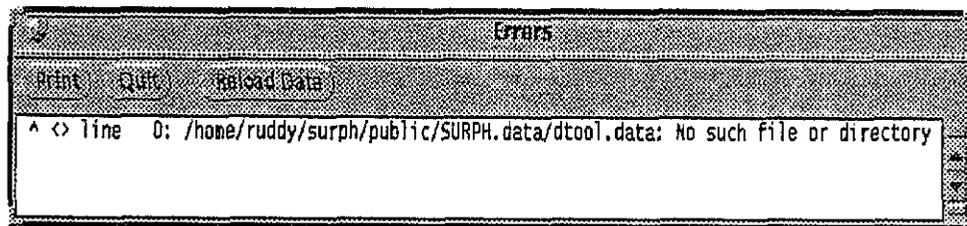
9.2 Description of the Tools Available in DataTool

This section will illustrate and briefly describe each window that the user will encounter while using DataTool. For more information on the use of DataTool, see Section 9.3.

To begin DataTools, at the prompt, type "datatool filename" where filename is the name of a SURPH-readable datafile. For the purposes of illustration we will use the datafile "dtool.dat"

```
> datatool dtool.dat
```

If you attempt to access a file that is either not SURPH-readable or does not exist, the following dialog-box will appear.

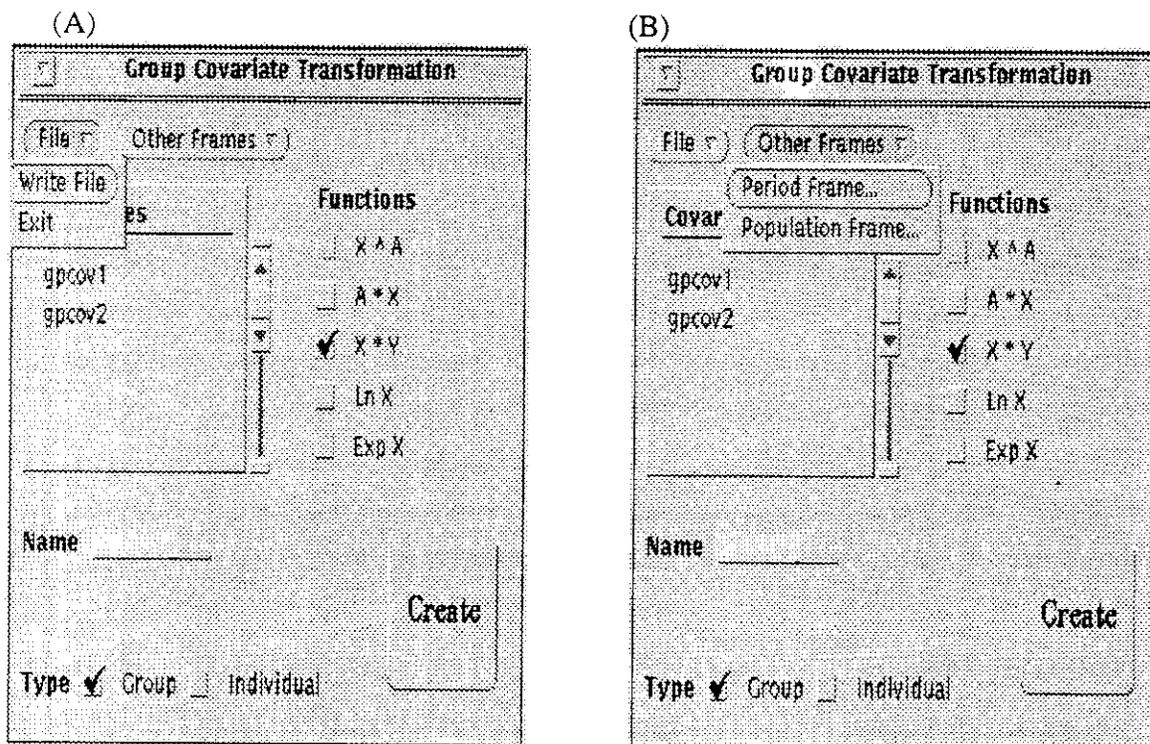


In the above example, the user attempted to access the file "dtool.data". An Error Window informs the user that the file "dtool.data" does not exist.

9.2.1 Group-Covariate Transformations

The initial window is the Group Covariate Transformation Window. The Group Covariate Transformation Window is used to generate new Group Covariates that are functions of the currently defined Group Covariates. As an example, suppose a group covariate in the data set was the concentration of **ATPase**, denoted “**ATPase**”. The user may wish to create a new covariate which is the natural log of the covariate “**ATPase**”. To create the new group covariate, the user could use the Group Covariate **Transformation** frame of DataTool

On the left-hand side of the window, the group **covariates are** listed by name. On the **right-** hand side of the window, the transformations **are** listed. The (A) File Button at the top of the window allows the user to Write the transformed data to a new SURPH-readable file and to Quit **DataTool**. The (B) Other Frames Button **allows** the user to move to the other types of transformations that **are** available (i.e., period and population transformations).

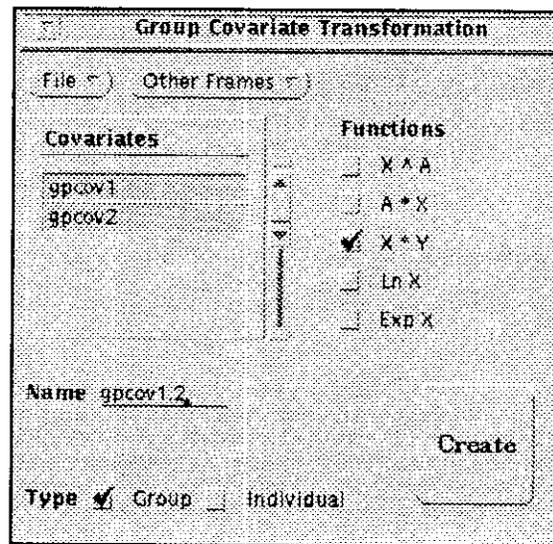


from the Covariates List and the functional transformation from the Functions List. To name the

new Covariate, type the name you desire in the blank line following “Name” prior to pushing the Create Button. If you forget to include a name for the new variable, a pop-up window **will** appear and advise you to name the new variable. You cannot create a new variable without naming it. To create the new **Covariate**, place the arrow key above the Create Button, then left-click.

Available functional manipulations are:

- X^A** **Raise** the value of the selected Covariate (**X**) to the power “A”.
- A*X** Multiply the value of the selected Covariate (X) by “A”.
- X*Y** Multiply the value of one selected Covariate (X) by the value of another selected Covariate (**Y**).
- Ln (X)** Take the Natural Logarithm of the **value** of the selected Covariate (X).
- Exp (X)** Exponentiate the value of the selected Covariate (X).

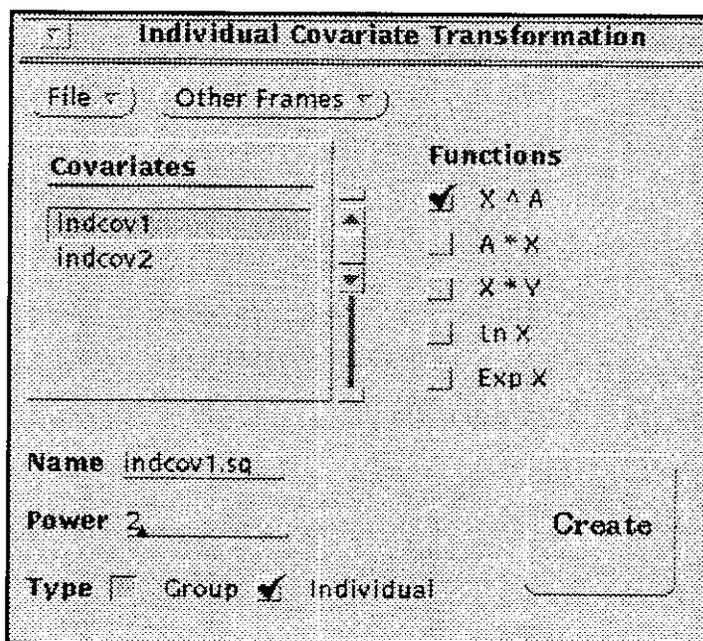


In the example window, left-clicking on the Create Button will create a new variable “*gpcov1.2*”. The values of *gpcov1.2* for each population will be equal to the values of *gpcov1* multiplied by *gpcov2*.

9.2.2 Individual-Covariate Transformations

The Individual Covariate Transformations allows the user to combine or alter the values of individual covariates that were collected. For example, suppose the user collected data on length, width and height of an object. If the object of interest is shaped like a cube, a variable of interest may be “Volume”, **which** could be approximated using “length x width x height”. To create the new covariate, the user could use the Individual-Covariate Transformation portion of DataTool

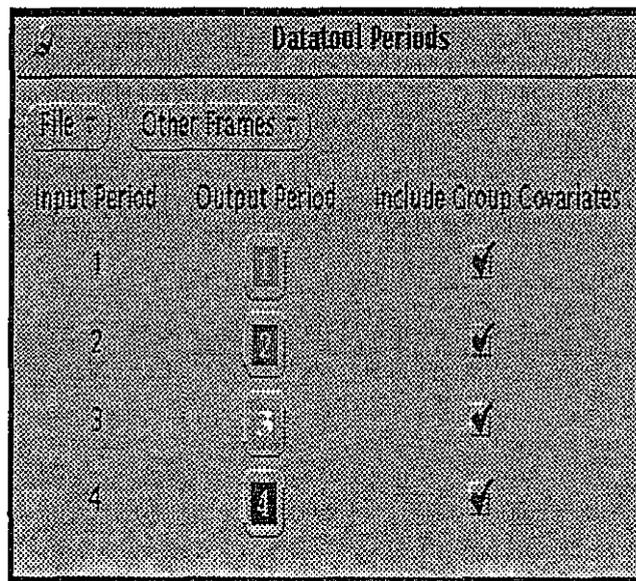
To get to the Individual Covariate Transform Window, left-click on the Individual **check-** box at the bottom of the Group Covariate Transform Window. The Individual Covariate Transform Window looks similar to and operates identically to the Group Covariate Transform Window. The sole difference between the two windows is that the objects of the transformations are the individual covariates rather than the group covariates.



In the example above, the values of *indcov1* will be transformed using the transformation function X^A , where “X” is the covariate that will be used (i.e., *indcov1*), and “A” is the power to which that covariate will be raised (i.e., 2). The name of the new variable will be *indcov1.sq*, and the values of the new variable will equal $(indcov1)^2$.

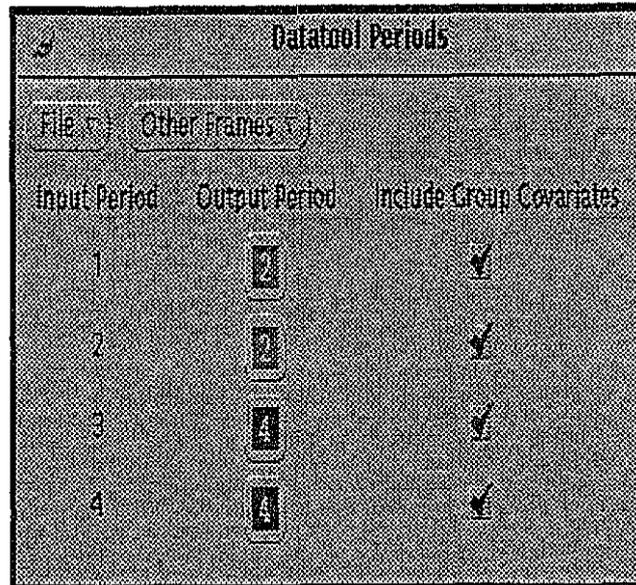
9.2.3 Period Transformations

The Periods Transformations Frame allows the user to combine periods over which the data was collected. For example, suppose the user collected data on a biweekly time-scale, but decided to analyze the data on a monthly time-scale. One way to collapse the data would be to assume that all animals that were released within a given month could be considered a single release. Similarly, **all** animals that were recaptured within a given month could be considered a single recapture. To collapse data in this manner the user can use the Data Tool Periods Frame (Other Frames -> Period Frame)



On the **far** left of the Datatool Periods window **are** the values of the Input Period Values. To the right of the Input Period Values are a set of buttons that may be changed to collapse the current periods into new period designations. The algorithm uses the logical "OR" to collapse periods. To the **far** right of the Datatool Periods window is a column of checkboxes. A checked box indicates that the collapsed period should include the group covariates. If the covariate is time-invariant, it will not matter whether the boxes are checked or not. If the covariate varies through time and the checkboxes are checked, Data Tool alters the values of the current group covariate to the average of the group covariate values for all those periods that are combined. If

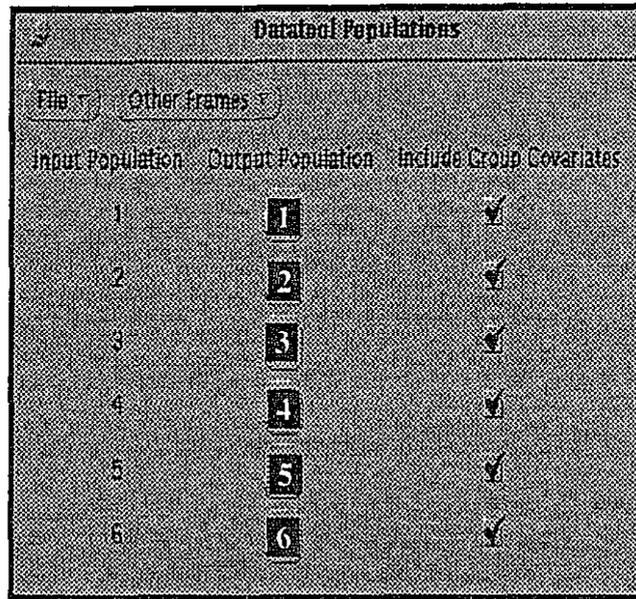
the **covariate** varies through time and the checkboxes **are** not checked, **DataTool** inserts zeros for all the group covariate values.



In the example above, periods 1 and 2 in the original dataset will be collapsed into a single period. Likewise, periods 3 and 4 in the original dataset will be collapsed into a single period. The checkboxes indicate that the new data will inherit the average values of the group covariates.

9.2.4 Population Transformations

The Population Transformations Frame allows the user to combine populations over which the data were collected. For example, suppose the user collected data from 2 blocks each within several different areas. One way to collapse the data would be to assume that both blocks within an area had the same survival, and thus constituted a single population, rather than two separate populations. To collapse data in this **manner**, the user can use the DataTool Populations Frame (Other Frames -> Population Frame).



On the **far** left of the Datatool Populations window are the values of the Input Population Values. To the right of the Input Population Values are a set of buttons which may be changed to collapse the current populations into new population designations. The algorithm uses the logical “OR” to collapse populations. To the far right of the Datatool Populations window is a column of checkboxes. A checked box indicates that the collapsed population should include the group covariates. If the covariate is time-invariant, it will not matter whether the boxes are checked. If the covariate varies through time and the checkboxes are checked, DataTool alters the values of the current group covariate to the average of the group covariate values for all those populations that are combined. If the covariate varies through time and the checkboxes are not checked, Data Tool inserts zeros for all the group covariate values.

Datatool Populations		
File	Other Frames	
Input Population	Output Population	Include Group Covariates
1	2	<input checked="" type="checkbox"/>
2	2	<input checked="" type="checkbox"/>
3	4	<input checked="" type="checkbox"/>
4	4	<input checked="" type="checkbox"/>
5	6	<input checked="" type="checkbox"/>
6	6	<input checked="" type="checkbox"/>

In the example above, populations 1 and 2 in the original dataset will be collapsed into a single population. Likewise, populations 3 and 4 in the original dataset will be collapsed into a second population, and populations 5 and 6 will be collapsed into a third population. The checkboxes indicate that the new data will inherit the average values of the group covariates.

9.2.5 Writing to a File

Once the transformations have been completed, the file must be saved before it can be used in SURPH. To save a file, use the command sequence File -> Write File. A pop-up window will appear that will allow the user to save the transformed datafile to a new name.

Write Data	
Directory:	/mounts/ruddy/surph/public
File:	dtool.b.dat
Cancel	Write

To specify the directory where the datafile should be saved, left-click on the Directory field and type in the directory path. To specify the name of the new datafile, left-click on the File field, and type the name of the data file. Once the correct directory has been specified and the file name has been entered **into** the field, left-click on the Write Button. In this example, the model file “**dtool.b.dat**” would be created **in** the directory “**/mounts/ruddy/surph/public**”.

9.3 Examples of Using DataTool

This section will be used to illustrate the concepts that were presented in Section 9.2. A **dataset** specifically created for this section (i.e., **dtool.dat**) will be used to illustrate the various procedures and pit-falls.

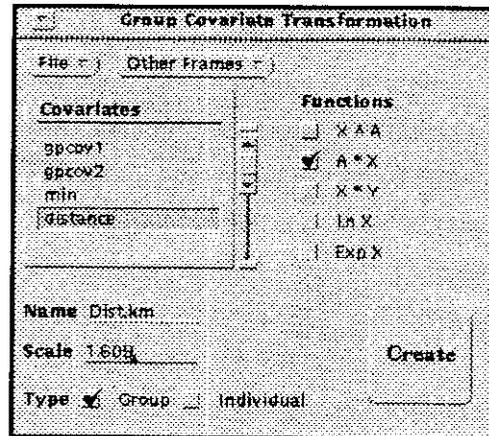
9.3.1 Group-Covariate Transforms

Suppose that the user originally collected data on the miles covered by an group of animals during a yearly migration (i.e., “distance”) and expected that **variable** to **be related** to survival. However, after the SURPH-readable datafile had been constructed, the user decided that(s)he wanted to use kilometers as the independent variable instead of miles.

To begin DataTools, at the prompt, type “datatool filename”, where filename is the name of a SURPH-readable datafile. For the purposes of illustration, this example will use the datafile “dtool.dat”

```
> datatool dtool.dat
```

To create a new group-covariate variable that has values equal to a scalar of the original group covariate, select the current group-covariate variable from the list (i.e. “distance”). Check the A*X transform, and name the new variable “Dist.Km”. Put the conversion factor for miles to kilometers (i.e., 1.609) in the scale field. Left-click OR Create. Save the new datafile under dtool.b.dat (File-> Write).



The new datafile now has a new group covariate called “Dist.Km” (Figure 9.1). Otherwise, except for some minor cosmetic differences, the new datafile is exactly like the old datafile.

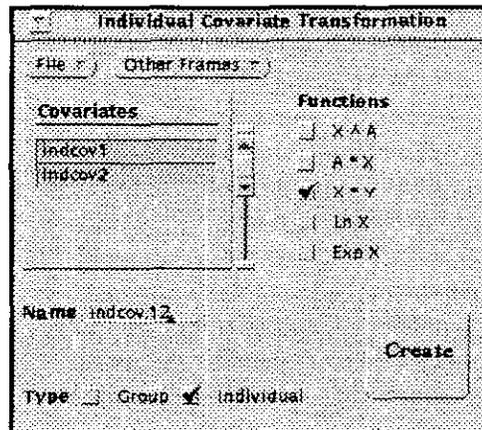
9.3.2 Individual-Covariate Transforms

Suppose that the user originally collected data on the length and girth of an animal. However, after the SURPH-readable datafile had been constructed, the user decided that he wanted to use a combined “length x girth” measure as the independent variable instead of either the length or girth *alone*.

Begin DataTools. Check the box labelled Individual at the bottom of the Group Covariate Transformation Window. To create a new individual-covariate variable that has values equal to the product of two other individual covariates, select the current individual-covariate variables from the list (i.e., “indcov1” and “indcov2”). Check the X*Y transform, and name the new variable “indcov.12”. Left-click on Create. Save the new datafile under “dtool.c.dat”.

(A)	(B)
Data Simulated telemetry data from telem_sim_win	Data Simulated telemetry data from telem_sim_win
npop 6 nper 4	number-of-populations 6 number-of-periods 4
gcv name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000	group-covariate name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
gcv name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000	group_covariate name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000
gcv name min time 1 19 4 13 -7 19 4 13 -7	group-covariate name min time 1 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000
gcv name distance time 0 116 176 235 345 467 748	group-covariate name distance time 0 116.000000 176.000000 235.000000 345.000000 467.000000 748.000000
icov indcov1 icov indcov2	group-covariate name Dist.km time 0 166.644000 283.184000 378.115000 555.105000 751.403000 1203.532000
ntag 50 30 0 0 50 30 0 0 50 30 0 0	

Figure 9.1 Comparison of the (A) original datafile (Le., dtool.dat) to the (B) transformed datafile (i.e., dtool.b.dat).



The new datafile now has a new individual covariate called “**indcov.12**” (Figure 9.2) which is the product of the values of “indcov1” and “indcov2” for each individual (e.g., **for the first** individual, “indcov1” = 32.023, “indcov2” = 4.214, and “**indcov.12**” = $32.023 * 4.214 = 134.954$). Otherwise, except for some minor cosmetic differences, the new datafile is exactly like the old datafile.

9.3.3 Period Transforms

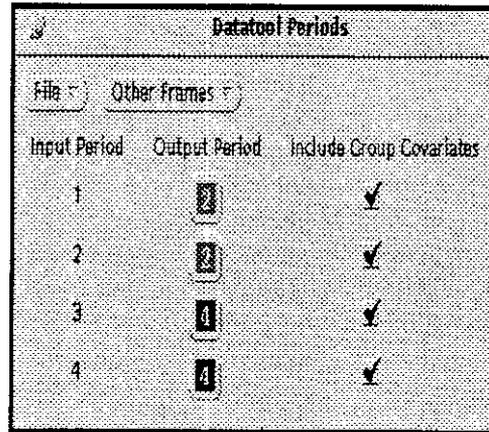
Suppose that the user originally collected data on a biweekly time-scale. However, **after** the SURPH-readable datafile had been constructed, the user decided that **(s)he** wanted to use a monthly time-scale as the interval of interest. To do so, the data from the biweekly collections must be combined in some coherent fashion. The program Datafile combines periods by using the logical “OR”.

Begin DataTools. Go to the Period Frame (Other Frames -> Period Frame). To create a new time-scale one that combines the biweekly measures into monthly measures, change the buttons under Output Period. The first two Input Periods will correspond to the first Output Period, so change the buttons from “1” and “2” to “2” and “2”. Similarly, the last two Input Periods will correspond to the second Output Period.

(A)	(B)
<pre> Data Simulated telemetry data from telem_sim_win npop 6 nper 4 gcov name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 gcov name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000 gcov name min time 1 19 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 gcov name distance time 0 116 176 235 345 467 748 icov indcov1 icov indcov2 ntaa 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 idien 0 data 1 1 1 0 0 32.023027 4.214295 1 1 1 1 0 43.662921 4.081977 1 1 0 0 0 16.680651 5.554117 1 1 0 0 0 57.923423 4.608262 1 1 1 0 0 22.505242 3.736325 1 1 0 0 0 64.024929 6.446260 </pre>	<pre> Data Simulated telemetry data from telem_sim_win number-of-populations 6 number-of-periods 4 group-covariate name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000 group-covariate name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000 group-covariate name min time 1 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 19.000000 4.000000 13.000000 -7.000000 group-covariate name distance time 0 116.000000 176.000000 235.000000 345.000000 467.000000 748.000000 individual-covariate indcov1 individual-covariate indcov2 individual-covariate indcov.12 number-taaad 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 50 30 0 0 length_of_ids 0 full_hist data # population 1 1 1 1 0 0 32.023 4.21429 134.954 1 1 1 1 0 43.6629 4.08198 178.231 </pre>

Figure 9.2 Comparison of the (A) original **datafile** (i.e., **dtool.dat**) to the (B) transformed **datafile** (i.e., **dtool.c.dat**).

Change the buttons from “3” and “4” to “1” and “2”. Leave all the checkboxes checked. Save the new datafile under “dtool.d.dat”.



The new datafile (i.e., “**dtool.d.dat**”) has been changed in three places due to the transformation (Figure 9.3). The most obvious change is the reduction in periods. As stated before, the transformation uses the logical “OR” to combine periods (e.g., “1” OR “1” = “1” “0” or “1” = “1”, “1” OR “0” = “1” “0” or “0” = “0”). Thus, any individual that was released in either the first or second periods would get a “1”, and similarly, any individual that was recaptured in either the third or fourth periods would get a “1”. The second place the datafile has been changed is in the “**number_tagged**”. Notice that since the first and second release periods were combined, the number tagged was summed (e.g., 50 + 30 = **80**), and recorded as released in the first period. The third place the datafile has been changed resulted in some unforeseen consequences. Notice that the **time-varying** group-covariate “**min**” has been changed. Initially (Figure 9.3 (A)), this variable **was** the minimum temperature during the biweekly interval. However, after the transformation (Figure 9.3(B)), Data Tool has replaced the minimum temperature with the average minimum temperature for the monthly time-scale. This transformation is probably not desired, and the user must go in and change this before further analyses **are** conducted.

9.3.4 Population Transforms

Suppose that the user originally collected data on 6 populations, (e.g., 3 Blocks x 2 Treatments) but after the original analysis, (s)he decided **that** all three blocks behaved similarly.

(A)

(B)

(A)	(B)
Data Simulated telemetry data from telem_sim_win	Data Simulated telemetry data from telem_sim_win
npop 6 nper 4	number_of_populations 6 number-of-periods 2
gpcov name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000	group_covariate name gpcov1 time 0 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
gpcov name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000	group_covariate name gpcov2 time 0 0.000000 1.000000 2.000000 0.000000 1.000000 2.000000
gpcov name min time 1 18 4 13 -7 19 4 13 -7 19 4 13 -7 19 4 13 -7 13 4 13 -7	group_covariate name min time 1 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000 11.500000 3.000000
gpcov name distance time 0 116 176 235 345 467 748	group_covariate name distance time 0 116.000000 176.000000 235.000000 345.000000 467.000000 748.000000
icov indcov1 icov indcov2	individual-covariate indcov1 individual-covariate indcov2
ntag 50 30 0 0 50 30 0 0	number_tagged 80 0 80 0 80 0 80 0 80 0 80 0 80 0 80 0
idlen 0 data 1 1 1 0 0 32.023027 4.214295 1 1 1 1 0 43.662921 4.081977 1 1 0 0 0 18.680851 5.554117 1 1 0 0 0 57.923423 4.608262 1 1 1 0 0 22.505242 3.738325 1 1 0 0 0 64.024929 6.446260 1 1 1 1 0 62.999350 5.897147 1 1 0 0 0 60.674568 5.107279 1 1 1 1 0 60.303010 4.443959	length_of_ids 0 full_hist data # population 1 1 1 0 32.023 4.21429 1 1 0 43.6628 4.08198 1 0 0 18.6809 5.55412 1 0 0 57.9234 4.60826 1 1 0 22.5052 3.73833 1 0 0 64.0243 6.44626

Figure 9.3 Comparison of the (A) original datafile (i.e., dtool.dat) to the (B) transformed datafile (i.e., dtool.d.dat).

Thus, to increase the power of the calculations, the user may wish to collapse the current **6-population** design into a **2-population** design. Note, however, that subsequent analyses ignore any site-to-site differences in survival (i.e., σ_s^2) probabilities, and treatment comparisons are based on measurement error alone. To collapse the current **6-population** design into a **2-population** design, the user could use the Population Transformation Frame (Other Frames -> Population Frame). Assume that the variable named “**gpcov1**” indicates the Treatments and the variable “**gpcov2**” indicates the Blocks.

Begin DataTools. Go to the Population Frame (Other Frames -> Population Frame). To create the new population designations, one that combines the three blocks into a single block, change the buttons under Output Period. The first three Input Populations will correspond to the first Output Population, so change the buttons from “1” , “2” and “3” to “3” , “3” and “3”. Similarly, the last three Input Populations will correspond to the second Output Population. Change the buttons from “4” , “5” and “6” to “6” , “6” and “6”. Leave all the checkboxes checked. Save the new datafile under “dtool.e.dat”.

The screenshot shows a dialog box titled "Datatool Populations" with a menu bar containing "File" and "Other Frames". Below the menu bar is a table with three columns: "Input Population", "Output Population", and "Include Group Covariates". The table contains six rows of data. In the "Output Population" column, the first three rows have the value "3" and the last three rows have the value "6". All checkboxes in the "Include Group Covariates" column are checked.

Input Population	Output Population	Include Group Covariates
1	3	<input checked="" type="checkbox"/>
2	3	<input checked="" type="checkbox"/>
3	3	<input checked="" type="checkbox"/>
4	6	<input checked="" type="checkbox"/>
5	6	<input checked="" type="checkbox"/>
6	6	<input checked="" type="checkbox"/>

The new datafile (i.e., “**dtool.e.dat**”) has been changed in several places due to the transformation (Figure 9.4). The most obvious change is the reduction in populations (i.e., from 6 to 2). As stated before, the transformation uses the logical “OR” to combine populations. Thus, any individual that was released as part of either the first, second or third original populations would be combined to form the first population in the new data, and similarly, any individual that was released as part of either the fourth, fifth or sixth original populations would be combined to form the second population in the new data. A second place the datafile has been

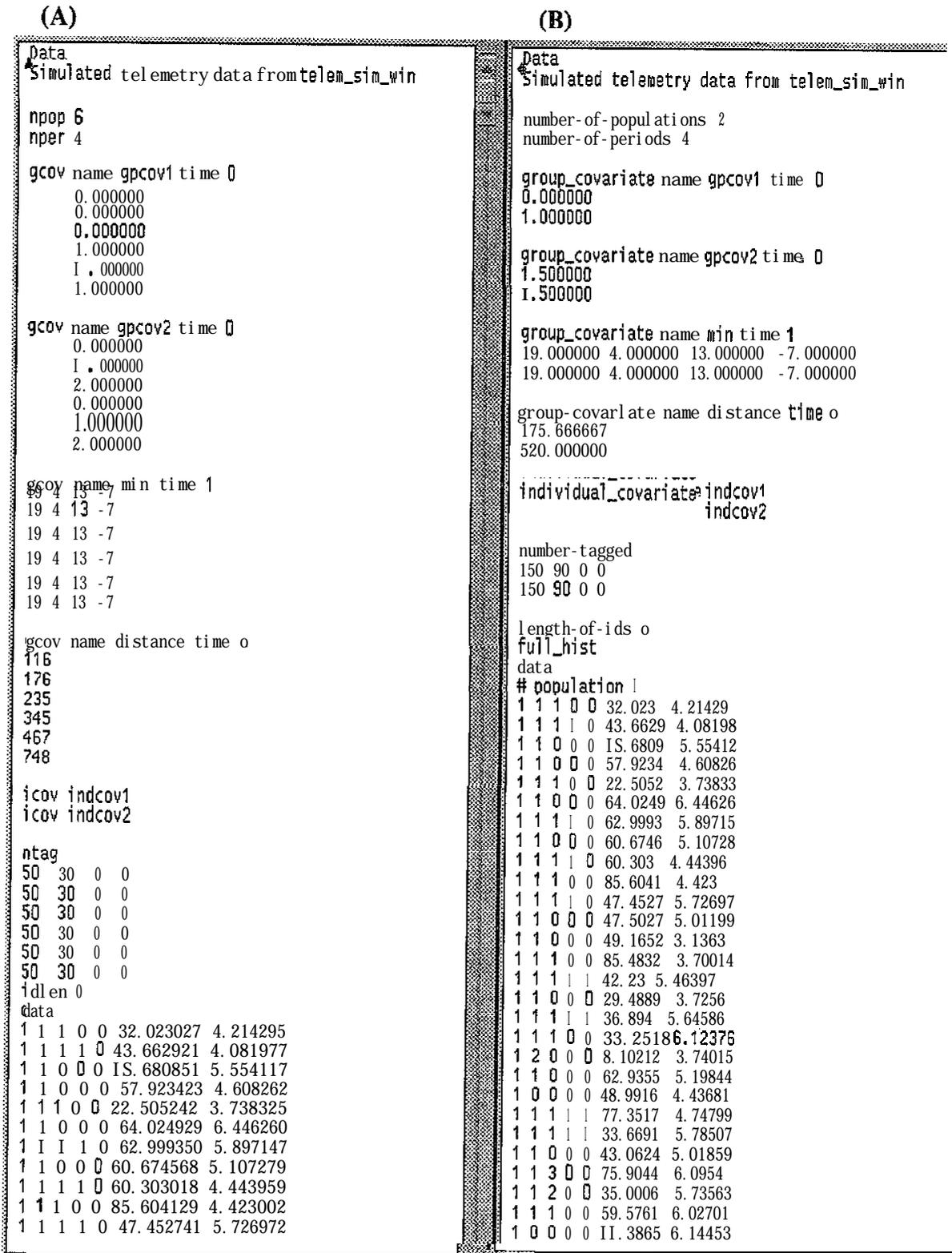


Figure 9.4 Comparison of the (A) original **datafile** (i.e., **dtool.dat**) to the (B) transformed **datafile** (i.e., **dtool.e.dat**).

changed is in the **“number_tagged”**. Notice that since the first three populations were combined, the number tagged was summed (e.g., $50 + 50 + 50 = 150$ for the **first** period, and 90 for the second period), and recorded as released as the first population.

Notice that because the group-covariate checkboxes were checked, all of the **group** covariates have been averaged across the populations. Thus, for the **first** group covariate, the first three original populations were from 1 treatment group, and **all** had a “0”. The average of the three “0’s” is a zero, so this is desired. Similarly, the second group covariate was the covariate we over which we combined populations (e.g., Blocks). We had already assumed that there was no effect of the second covariate (e.g., Blocks), and the identical values of 1.5 indicate that the populations have identical effects of **“gpcov2”**. The effect of the transformation on the group covariate “min” was to average the value across populations, and since all populations shared the same values per period, the only effect was to reduce the number of lines from **“6”** to **“2”**. Finally, the group covariate “distance” has been averaged. The user must decide whether this variable is now useful.

These examples have illustrated the use of DataTool. Although each example only used a single transformation, it is possible to combine periods, populations, group and individual covariates concurrently. However, as noted in the Period and Population Transformation Frames, sometimes the results from these transformations are not exactly what was desired. Therefore, the user may wish to limit the transformation to a single frame at a time. This will allow the user to verify more easily that the output datafile is desirable and sensible.

9.4 Conclusion

DatTool provides a coherent framework for transforming data from SURPH-readable files into new SURPH-readable files. DataTool transforms data at the individual and group-covariate level, and in addition, allows the user to combine data across periods and/or populations. This powerful tool should facilitate manipulations of datafiles used in SURPH.1.



Chapter 10

SURPN-PC Program

10.1 Introduction

Surph-PC is the version of **SURPH.1** that **runs** under the **MS-Windows®** operating system. AU estimation and hypothesis-testing **algorithms** are identical to those used in the UNIX version. This chapter will familiarize the user with cosmetic and procedural differences present within the **MS-Windows®** operating system. This chapter does not replace Chapter 7 of the SURPH.1 manual. Instead, this chapter is intended to clarify Chapter 7 with respect to differences in **Surph-PC**. The page numbers that follow each new topic indicate where more detailed information may be obtained. The order of the chapter **will**, where possible, follow the order dictated by the buttons on the Surph-PC Base Window. For a full coverage of SURPH.1 (i.e., statistical properties, potential uses, examples, and functionality) please read Chapters **1-9** of the **SURPH.1** manual.'

10.2 Differences in Surph-PC

Base Window (pp. 7.27-7.30) -The Surph-PC Base Window (Figure **10.1a**) is cosmetically different from the UNIX version (Figure **10.1b**). To begin with, some buttons **are** labelled differently in the two versions. The Surph-PC labels and their UNIX analogues can be found in Table 10.1. Additionally, the upper portion of the Base Window looks quite different. In the UNIX version, the **ID**, Name, LnLik, # Parameters, S Link, P Link and AIC are displayed for both the model selected in the Master Model List, and the model selected in the Testable Model List. In Surph-PC, this information is displayed only for the model selected in the Master Model List.

Likelihood Ratio Tests (pp. 7.21-7.22) - In Surph-PC. the results of LRT **are** displayed in a small box on the lower right-hand portion of the Surph-PC Base Window.

Load Data, Load Models, Store Models (pp. 7.33-7.38) - As in the **UNIX** version of SURPH. 1, these functions reside on the File Pulldown Menu. However, whereas in the UNIX version the user is prompted to type the directory and filename in fields on a pop-up window, in

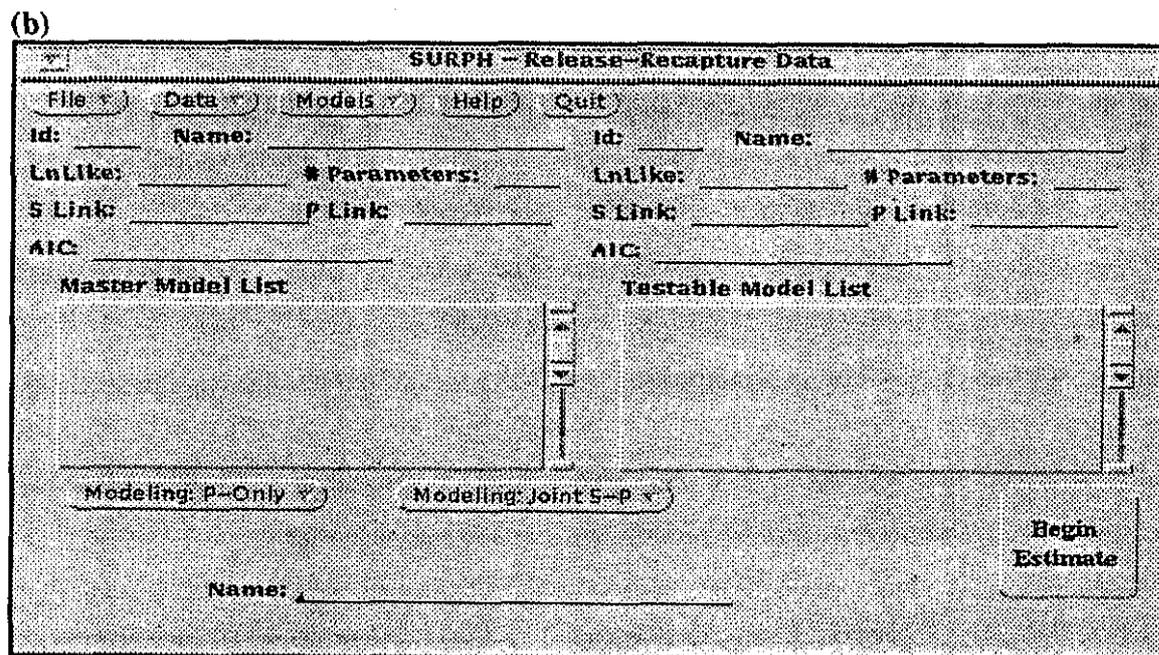
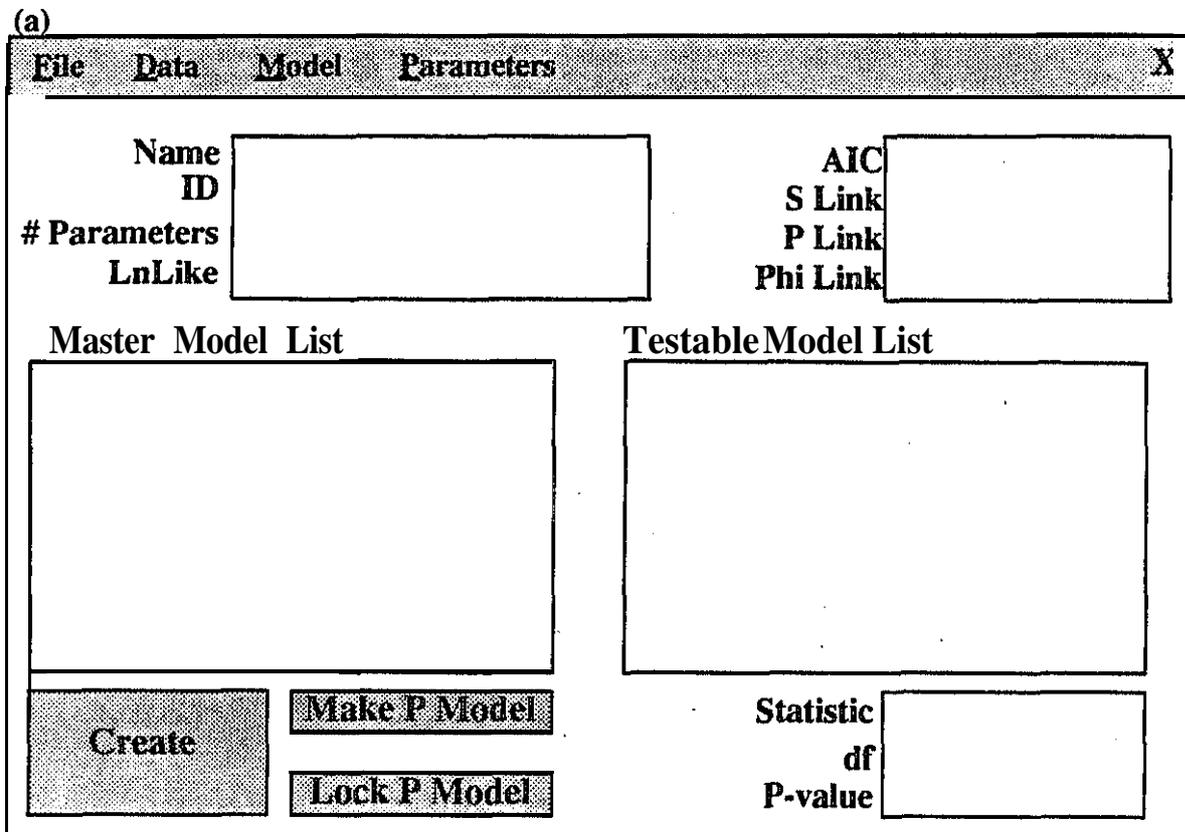


Figure 10.1 Surph Base Window for the (a) PC-Windows and the (b) Unix operating systems.

Table 10.1 Base-Window Button Labels. The button label from **the UNIX-version of SURPH** is given in the left-hand column, along with its PC analogue in the right-hand column.

UNIX	Surph-PC
File	File
Data	Data
Models	Parameters
Help	(Not Available)
Quit	Left-click on “Quit Window” icon. (X) in upper right comer.
Modeling: P-Only	Parameters
Lock	Lack P Model
Modeling: Joint S-P	Parameters
Begin Estimate	Create

Surph-PC, these operations are preformed through point-and-click mouse operations. During “load” and “store” operations, Surph-PC expects data files to use the **“.dat”** suffix, and model files to use the **“.mod”** suffix.

ANODEV (pp 7.111 - 7.122) - The ANODEV option resides on the File Pulldown Menu in Surph-PC. As with the Surph-PC Base Window, some buttons on the ANODEV Window have different labels in the Surph-PC version. The Surph-PC labels and their UNIX snalogues can be found in Table 10.2.

There are cosmetic changes to ANODEV as well. In **Surph-PC**, the ANODEV Window (Figure 10.2) has been split into two separate pop-up windows. The main window (Figure 10.2a) contains the information about the models that have been fit, the models that are used to compute the ANODEV Table, and shows the computed ANODEV Table. The second window (Figure 10.2b) is used to create and name additional models. In addition, quick-buttons have been added

Table 10.2 ANODEV Button Labels. The button label from the UNIX-version of SURPH is given in the left-hand **column**, along with its PC analogue in the right-hand column.

UNIX	Surph-PC
Calculate	Compute Table
Print	Print Table
Remove Model	Left-click on Transfer Bar
Transfer Model to Master List	Transfer Model
Quit	Left-click on "Quit Window" icon (X) in upper right corner.
Begin Estimation	Make Model

to the second window. In the LJNIX version of ANODEV (Figure 10.3), these windows are combined.

A procedural difference between Surph-PC and the **UNIX** version is how models are moved from the Available Models region to the Testable Models region. In Surph-PC, to move a model from one region to the other, select the model (left-click), then left-click on the transfer bars (i.e., ).

Individual Covariates (pp. 7.47-7.56) - As in the **UNIX** version of SURPH.1, this function resides on the Data Pulldown Menu. A histogram of the data or a cumulative distribution plot of the data may be displayed using this selection.

The windows that are displayed for the PC version have some minor cosmetic differences compared to their **UNIX** analogues (Figures 10.4 - 10.7). First, the positioning of **the** various components displayed has been changed. However, **all** options that are available **in the** UNIX version are available in Surph-PC. Second, the slide bar that was used to change the "Occasion" designation in the UNIX version has been replaced by a pull-down menu in Surph-PC. Finally, the

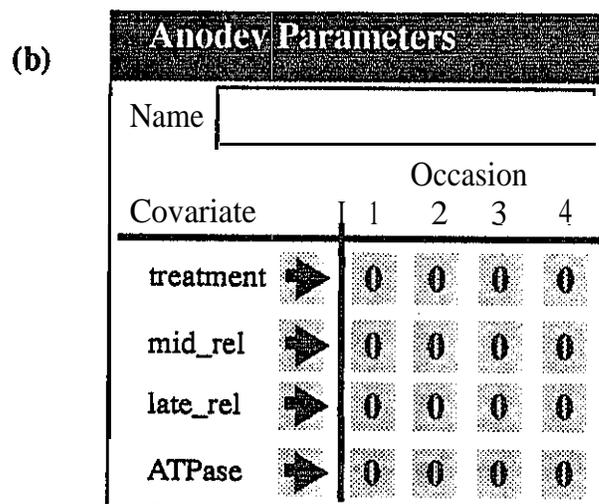
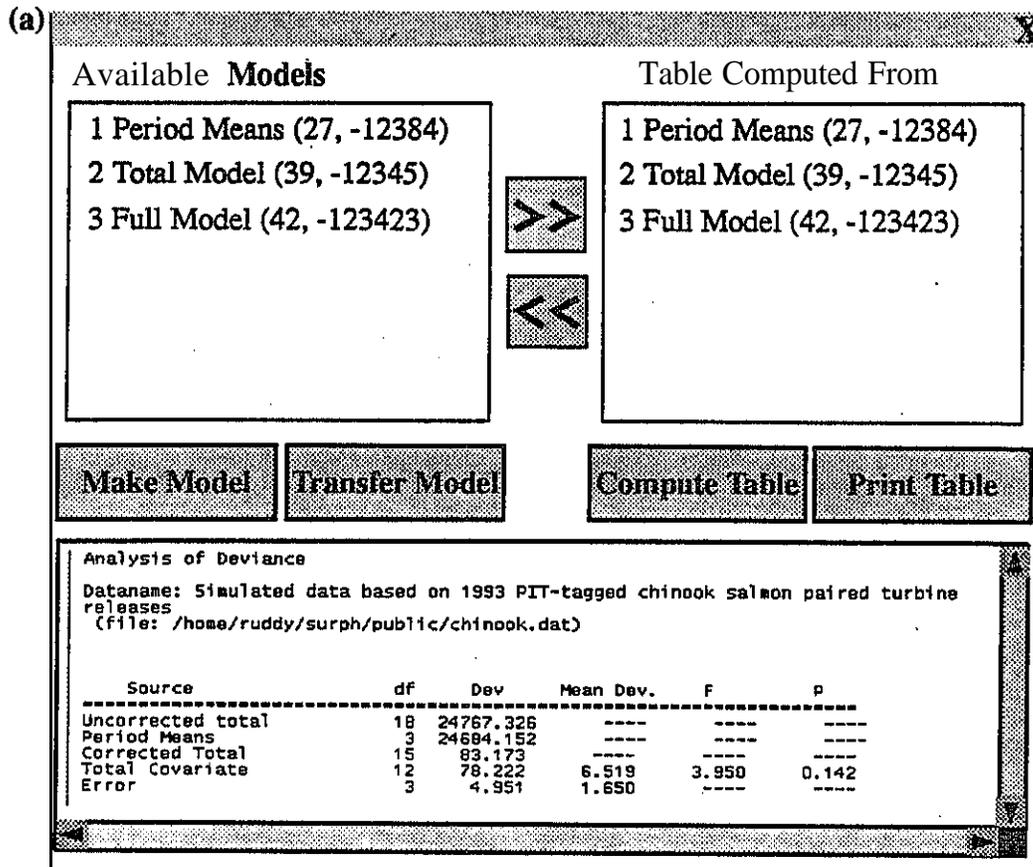


Figure 10.2 Surph-PC ANODEV Windows. The top figure (a) illustrates the ANODEV Table. Model names are displayed in the upper portion, whereas the ANODEV Table is displayed in the lower portion. The lower figure(b) is an ANODEV button pad. The ANODEV button pad is used to parameterize the group-covariate models.

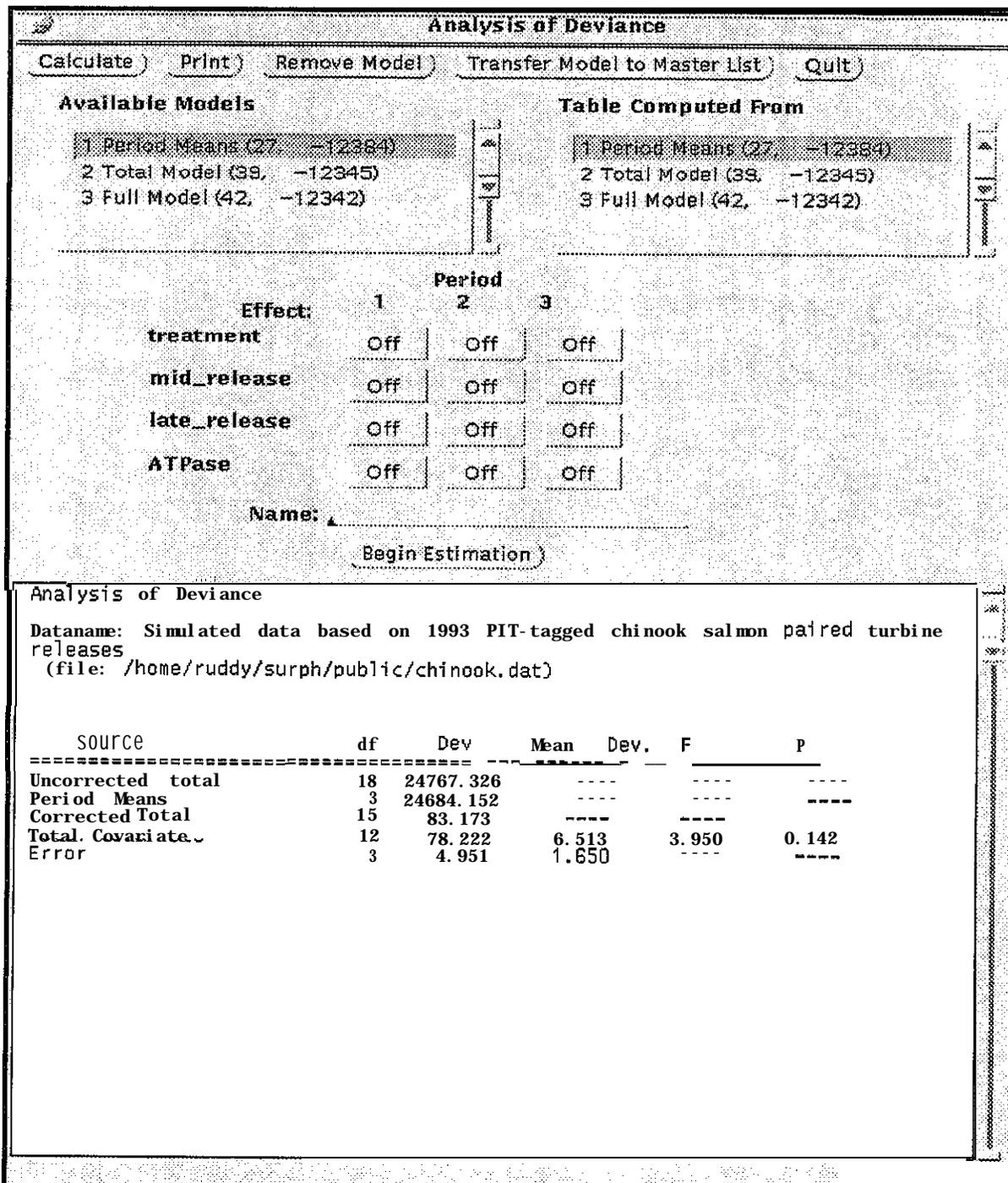


Figure 10.3 ANODEV Window from the Unix operating system. The major differences between the UNIX version and the PC version are that for the UNIX version (a) the ANODEV Button Pad is located within the ANODEV Window and (h) models are transferred differently.

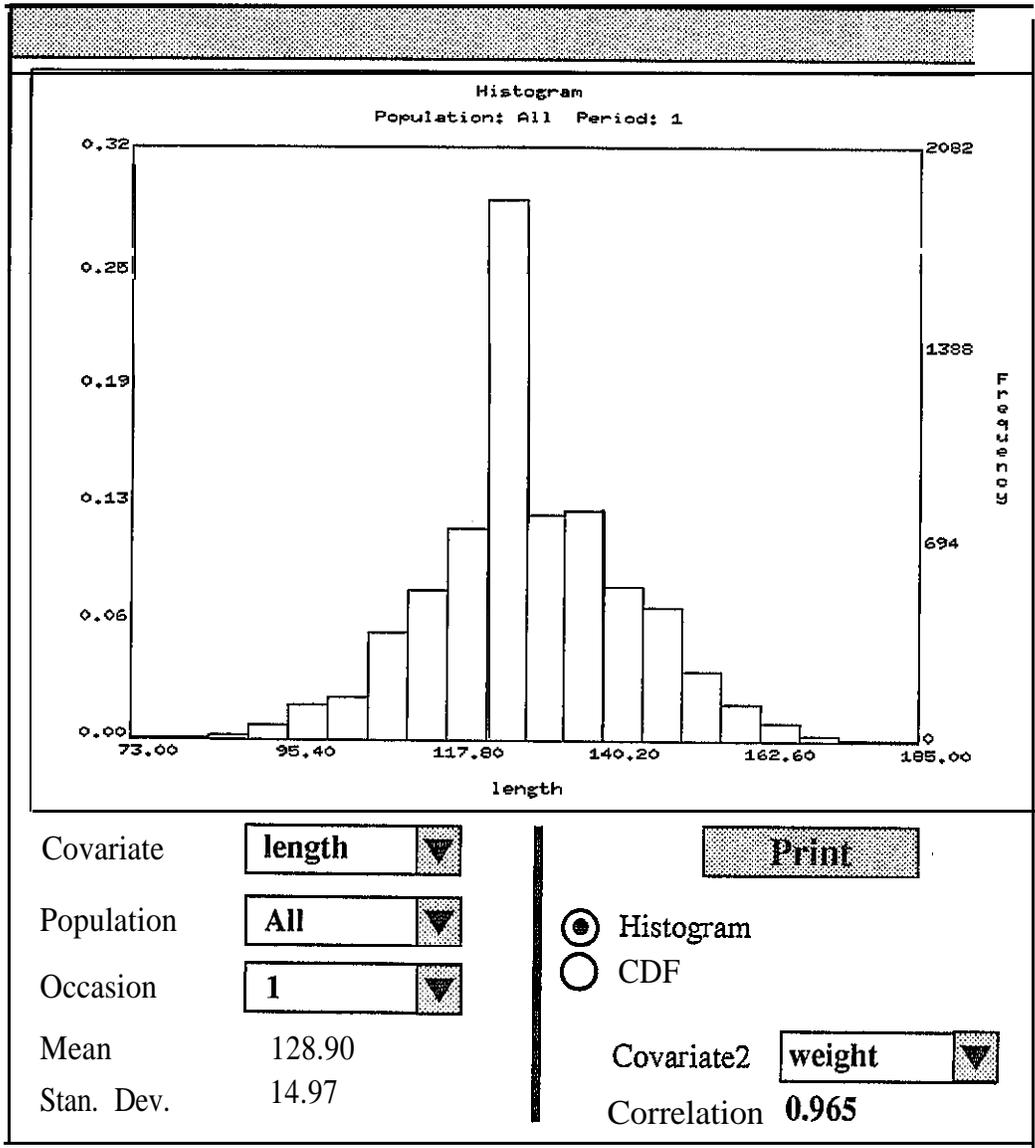


Figure 10.4 Individual Covariates Window for Snrph-PC - Histogram. A frequency histogram of the currently-selected covariate is displayed. On the left-hand side of the graphic, the relative frequency (probability) is given. On the right-hand side of the graphic, the frequency (count) is given. The horizontal axis denotes the value of the covariate.

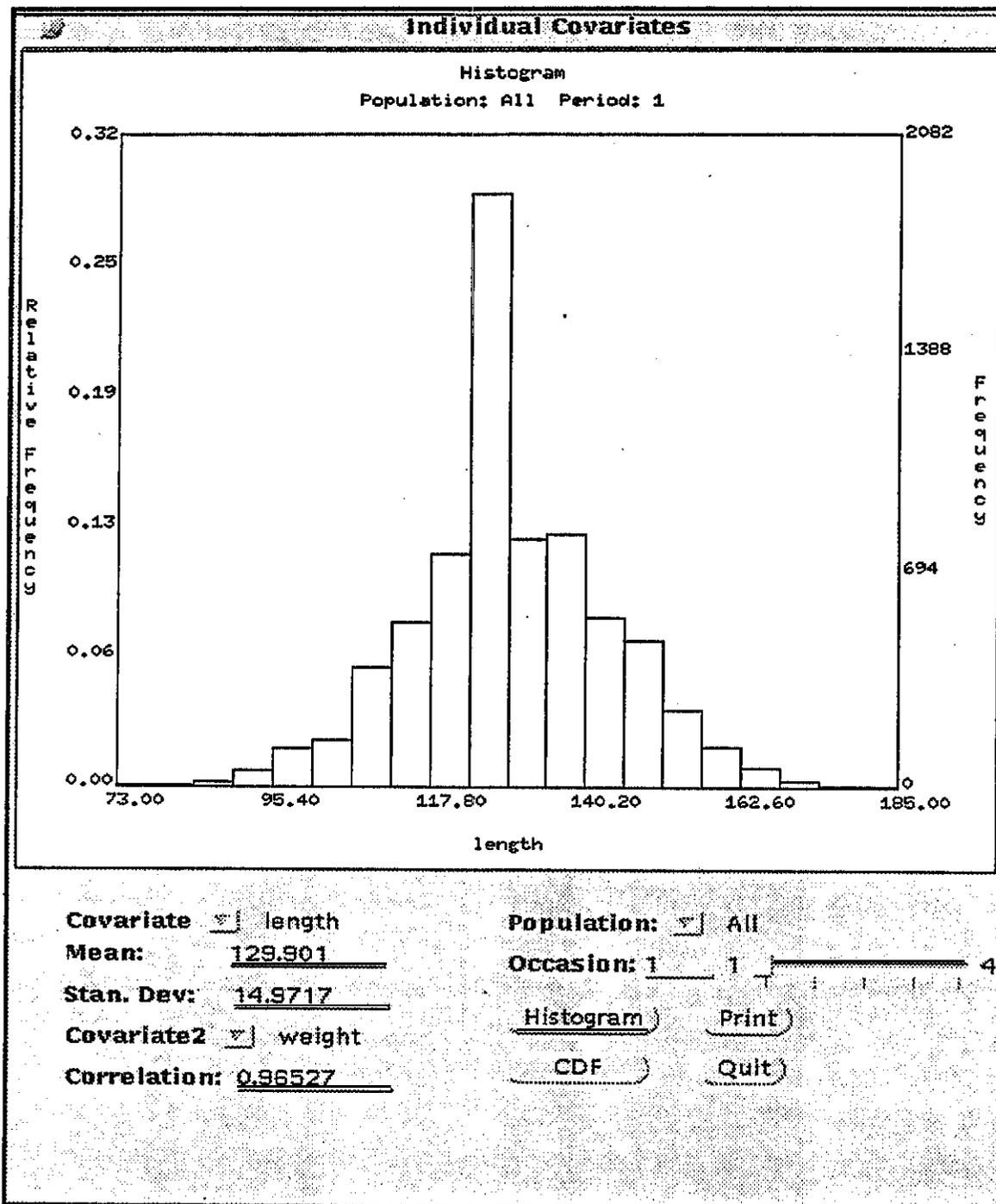


Figure 10.5 Individual Covariates Window for **UNIX SURPH**- Histogram. A frequency histogram of the currently-selected covariate is displayed. On the left-hand side of the graphic, the relative frequency (probability) is given. On the right-hand side of the graphic, the frequency (count) is given. The horizontal axis denotes the value of the covariate.

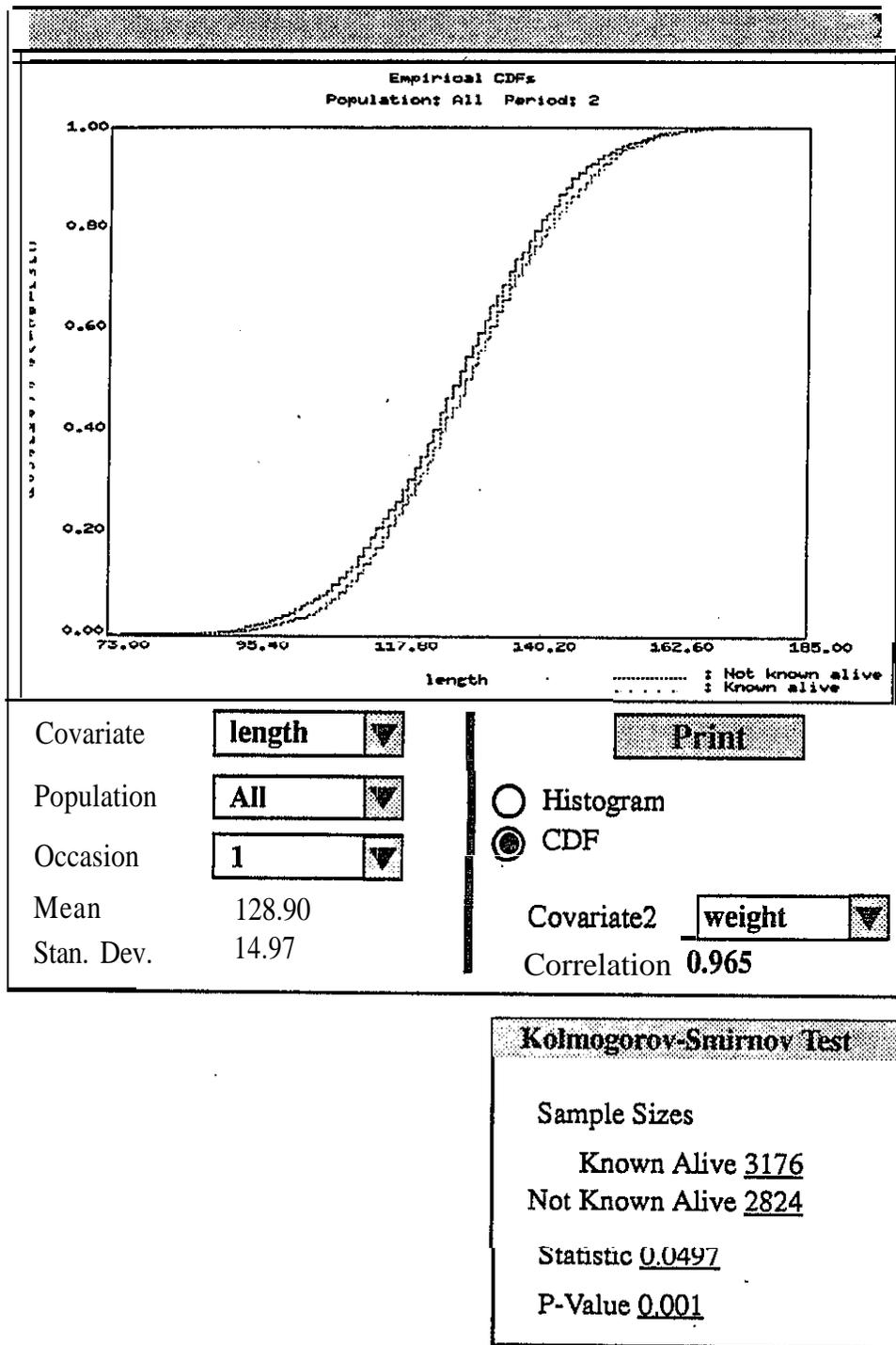


Figure 10.6 Individual Covariates Window for Surph-PC. Cumulative Distribution Plot and K-S test window are displayed. The Cumulative Distribution Plots for those animals “Known Alive” and those animals “Not Know Alive” are displayed, along with the results of a K-S test.

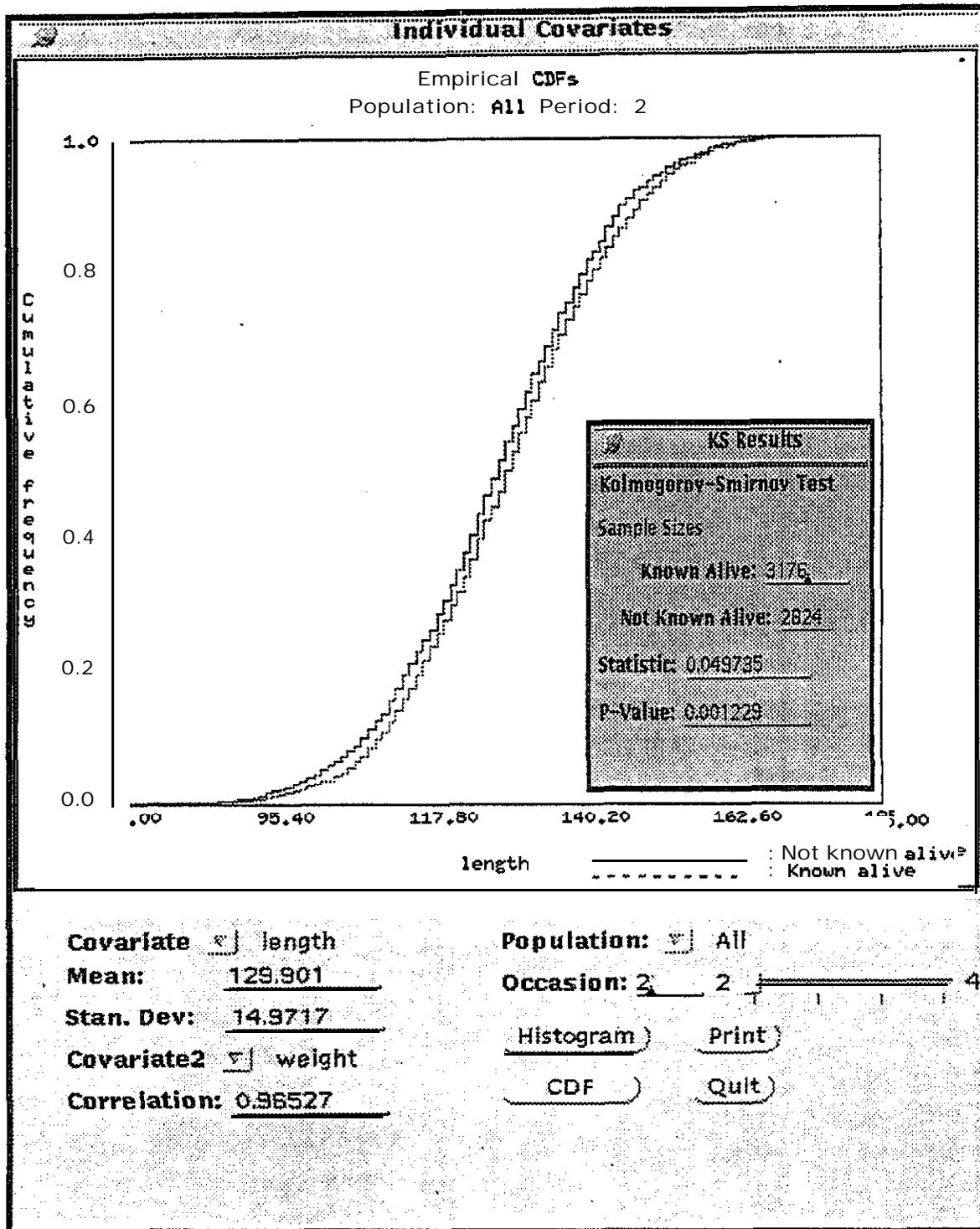


Figure 10.7 Individual Covariates Window for UNIX SURPH • Cumulative Distribution Plot and K-S test window. The Cumulative Distribution Plots for those animals “Known Alive” and those animals “Not Know Alive” are displayed, along with the results of a K-S test.

Quit Button has been removed. In Surph-PC the user must left-click on the “Quit Window” icon (i.e., X) in the upper right-hand corner to quit.

Capture Modeling (pp. 7.59-7.90) - One obvious **change** in Surph-PC is that all of the Button Pads are accessed using the Parameters PulldownMenu. Thus, in Surph-PC, the difference between P-Only Modeling and Joint S-P Modeling is defined by which button is depressed prior to estimation of the model (i.e., Create vs. Make P Model), not which Button Pads are accessed. To initiate P-Only Modeling in Surph-PC, the user accesses the Capture Button Pads from the Parameters Pulldown Menu. The user then selects the Make P Model button on the Surph-PC Base Window to estimate the model. In UNIX, to initiate P-Only modeling, the user would select the Capture Button Pads from the Modeling: P-Only PulldownMenu.

Renaming Models (pp. 7.173-7.174) - In the UNIX version, a Name field is provided on the SURPH Base Window prior to estimation. In Surph-PC, the name is initially fixed. The name can only be changed after estimation using the Rename command (Model -> Rename).

Survival, Capture and Product Curves (pp. 7.139-7.168) • As in the **UNIX** version of SURPH.1, these functions reside on the Model PulldownMenu. The graphics that are displayed (i.e., group and individual curves) have some minor cosmetic differences. Only the initial pop-up window is displayed for each **covariate** type (i.e., group or individual) as all the cosmetic changes have occurred to the pop-up window itself, not the graphic within the pop-up window.

For the window that displays the group-covariate curves (Figures 10.8a and 10.8b), the buttons have been moved. In Surph-PC, the buttons are located on the upper portion of the pop-up window, whereas in the Unix version of SURPH, the buttons are located on the lower portion of the pop-up window. Otherwise the pop-up window is almost identical.

For the window that displays the individual-covariate curves (Figures 10.9 and 10.10), there are only cosmetic differences. Primarily, the location of the buttons that select between the various SURPH-curve options has been **altered**. In Surph-PC, the buttons **are** located on the **left-**hand side of the window, and label is adjacent to the button. On the **UNIX** version, the label of the graphic to be displayed is atop the button. All of the buttons except for the Quit Button are

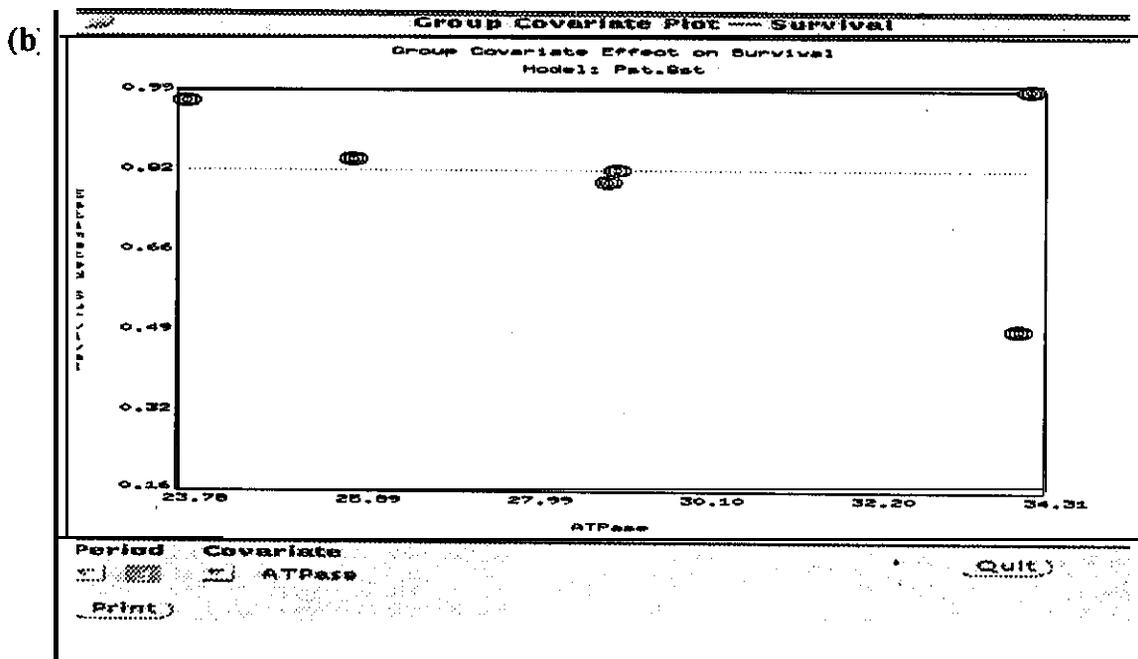
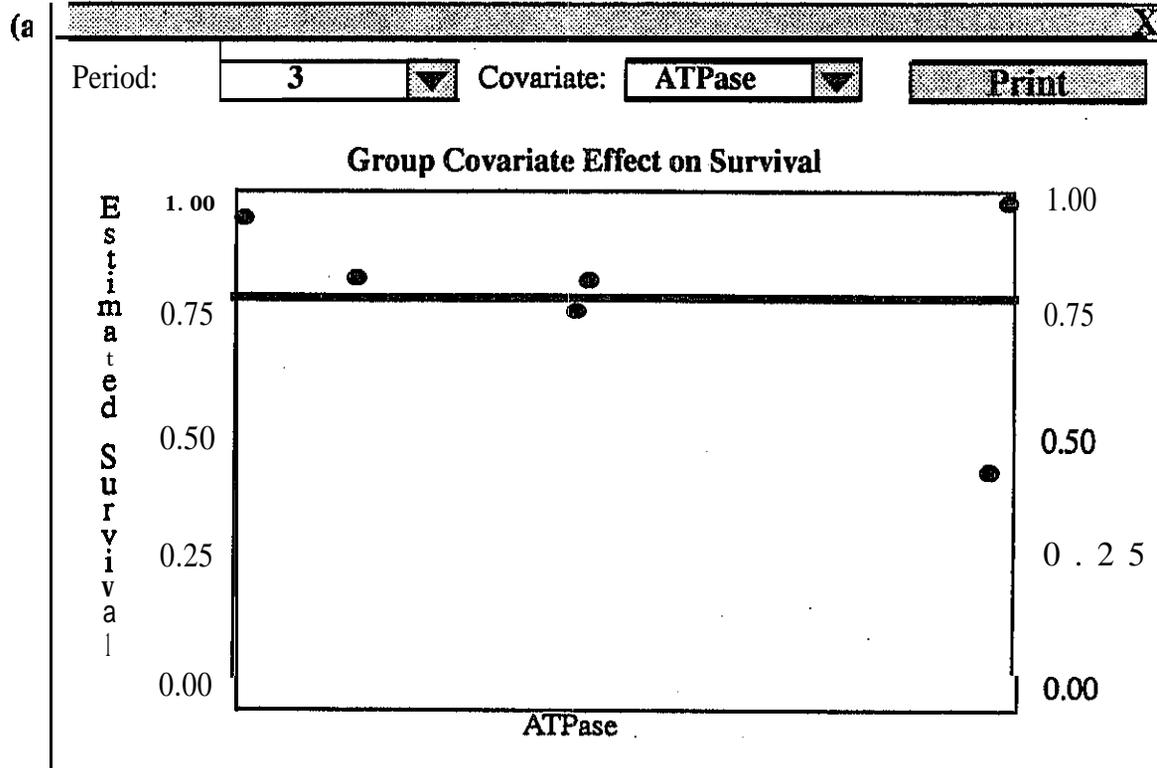


Figure 10.8 Model Graphics • Group Curves for (a) Surph-PC and for (b) the **UNIX** version. The vertical axis is the estimated survival. The horizontal axis spans the values of the group covariate. The fitted SURPH-model is displayed as a smooth curve. The **CJS** estimate of survival for each population is displayed as a point.

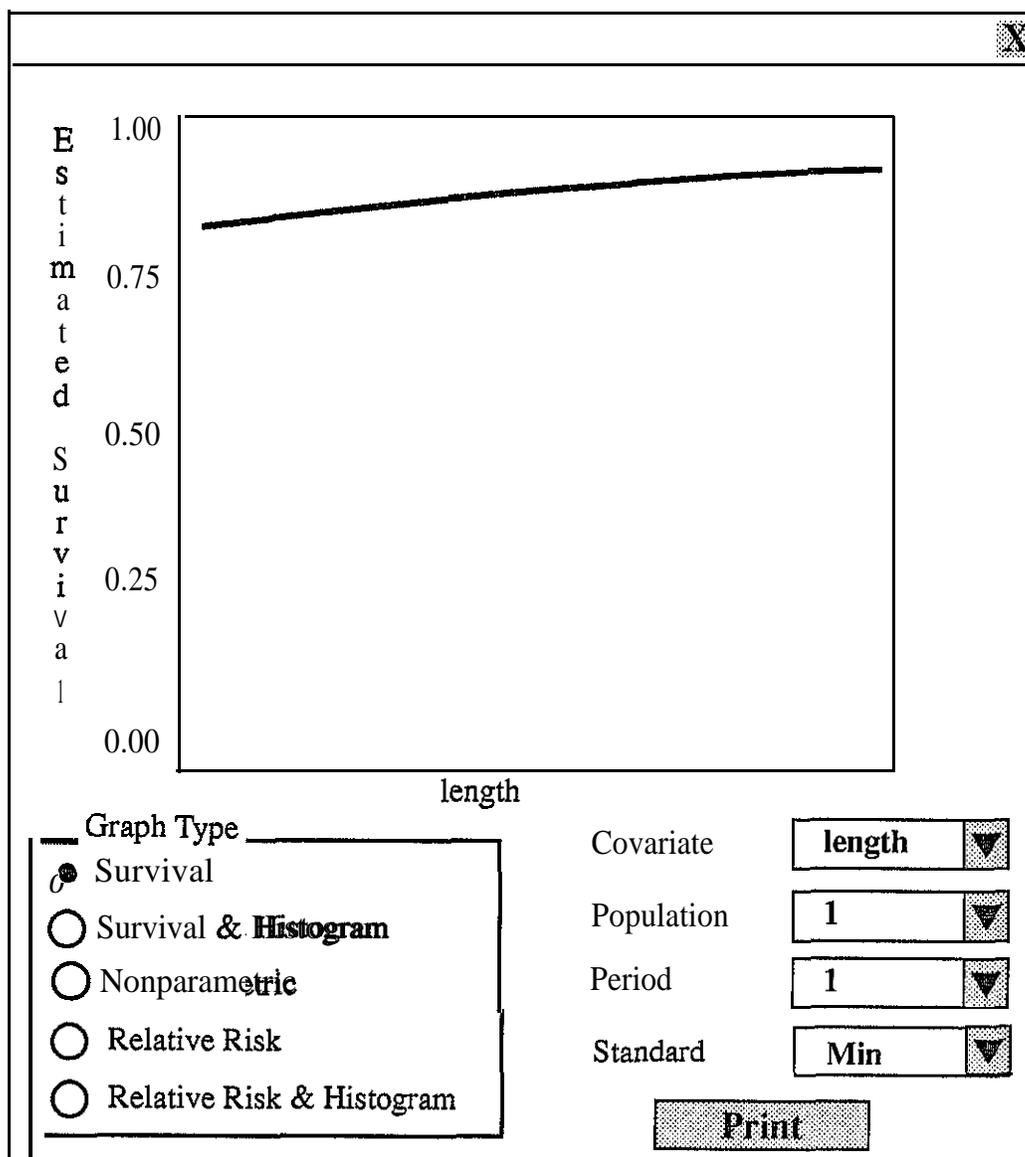


Figure 10.9 Models Graphics Individual SURPH Curves for Surph-PC. The vertical axis is the estimated survival. The horizontal axis spans the value of the covariate (i.e., length). This example graphic illustrates the curve obtained from the fitted SURPH-model. The user may also view the survival curve with the frequency histogram or with the nonparametric estimates of survival. Alternatively, the user may plot the relative risk curve, with or without the frequency histogram.

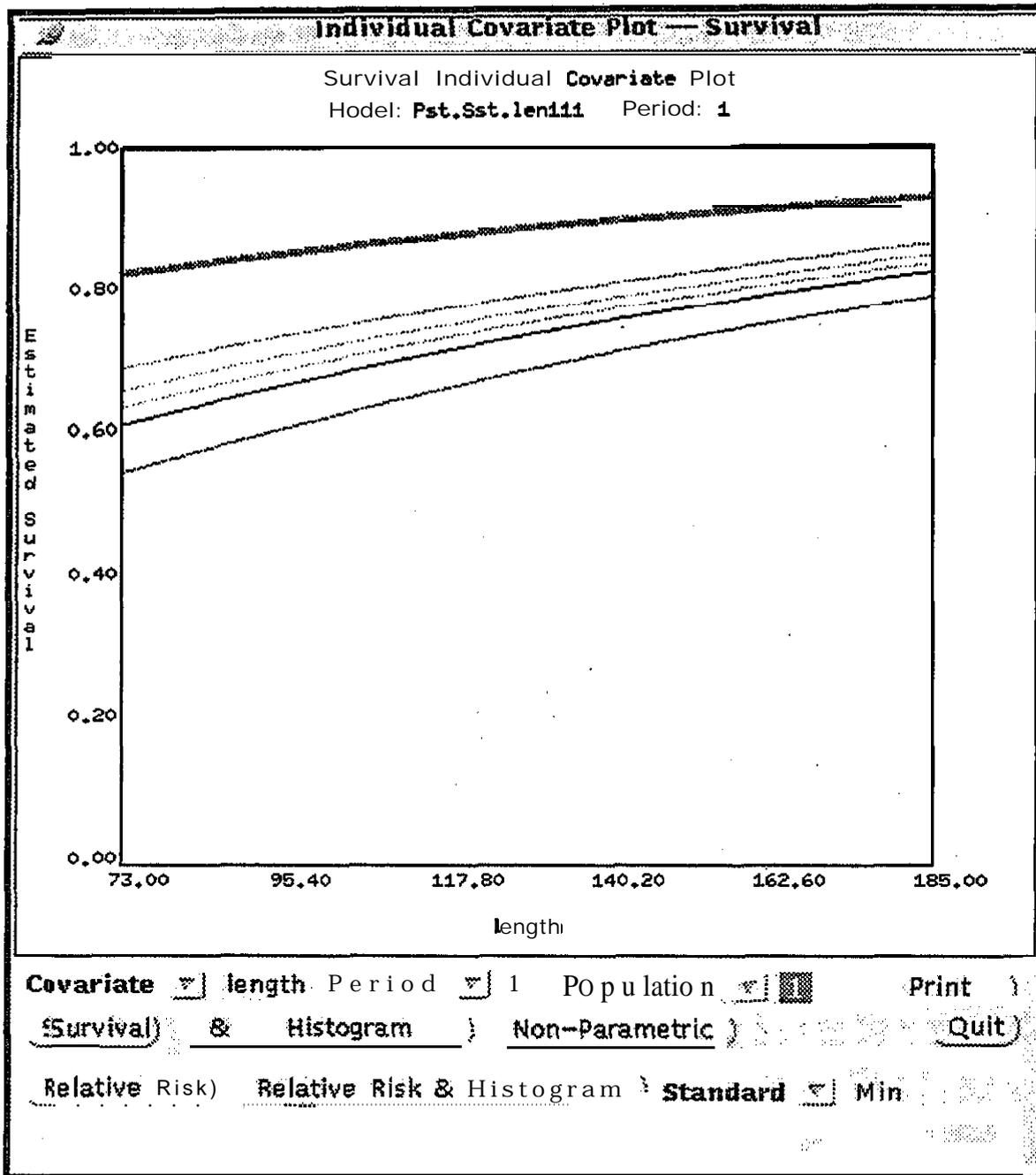


Figure 10.10 Models Graphics - Individual SURPH Curves for the UNIX version of SURPH. **The** vertical axis is the estimated survival. The horizontal axis spans the value of the covariate (i.e., length). This example graphic illustrates the curve obtained from the fitted SURPH-model. The user may also view the survival curve with the frequency histogram or with the nonparametric estimates of survival. Alternatively, the user may plot the relative risk curve, with or without the frequency histogram.

present. Similarly, the pulldown menus have been relocated. In Sutph-PC, all pulldown are located on the right-hand side of the window, whereas in the **UNIX** version, the pulldown menus were located above the graphics buttons.

Any other questions concerning **Surph-PC** should be clarified by reading Chapter 7 in the SLIRPH.1 manual.