1 The Lung Cancer data set (Kalbfleisch and Prentice, 1980)

1.1 The data

In this trial, men with advanced inoperable lung cancer were randomized to either a standard or test chemotherapy. The survival measure for therapy comparison is time from randomization to death. The information available in the file *lung.dat* is (in this order) :

1. *idnum*: an identity number
2. *days*: the time from randomization to death in days
3. *treat*: the type of therapy (1 = standard; 2 = test)
4. *cell*: the histological type of tumor (100 = squamous; 200 = small; 300 = adeno; 400 = large cell)
5. *code*: the censoring code (1= uncensored; 0 = censored)
6. *kps*: a measure, at randomization, of the patient's performance status or "Karnofsky rating" (10 -30 = completely hospitalized; 40 - 60 = partial confinement; 70 -90 = able to care for self) (continuous variable)
7. *age*: the age in years at randomization (continuous variable)
8. *diagtim*: time in months from diagnosis to randomization (continuous variable)
9. *prior*: a indicator of existence of prior therapy (0 = no; 10 = yes)

1.2 Analysis

A sequence of analyses is proposed in order to get familiar with the Survival Kit. All the indications below assume that the programs are run under Windows.

1. Getting the programs ready
   (a) Look at the *prepinclu.f* and *parinclu.f* parameter files (with the editor of your choice, e.g, *Wordpad, Notepad, Word, ...*.
   (b) From Survival Kit v6 onwards it is possible to modify the key parameters related to the database size in the parameter files for respective programs (i.e. *prepare.txt*, *cox.txt* and *weibull.txt*) without recompilation. If you want to compile the programs for some reason, a Fortran 90 compiler is needed.

If you compile the programs for large applications, it is **very important** to indicate that the compiled version of the program should be optimized (option -O3):

Drastic saving in computing time result from optimization. However, this is not so important in the examples treated here. Also, compiling the COX or WEIBULL programs with the -O3 option takes a few minutes, instead of a few seconds without optimization. See the 'User's Manual' for more notes on compilatin in Windows and Unix environments.

The *program executable* (e.g. *prepare.exe*) can be started via the command prompt.

In the Start menu (down and left), choose 'Run' and type 'cmd'. This will open an MSDOS window. The compiled executables and your data files should be in the same directory. Go there within the command prompt and type:

prepare.exe

to run the PREPARE program.

2. Using the program PREPARE

   (a) Look at the *prepare.txt* file in the LUNG directory. This is the parameter file which will be used as the input file for the PREPARE program.

   (b) Look at the original data set (*lung.dat*).

   (c) Run PREPARE (type: prepare.exe). The program will use the input parameters from the file *prepate.txt*. This has to be present in your directory, otherwise you get an error message (often: *The value of the STATUS specifier in an OPEN statement does not match the file status...* or similar).

   (d) Look at the output files from PREPARE:*codelist.txt ; varlist.txt; lung2.dat*.

3. Getting raw statistics using WEIBULL

   (a) Sorting: in this particular case (no stratification), there is no need to sort the data set in order to use WEIBULL.

   (b) Look at the parameter file *weibull.txt*. All covariates are listed and the statement ONLY_STATISTICS is used.

   (c) Run WEIBULL (type: weibull.exe). You have to have the file *varlist.txt* (coming from PREPARE) and the second parameter file for WEIBULL (*weibull.txt*) in your directory. The program will search for them automatically. In case they are missing, a similar error message will be displayed. If you don't use the LOGFILE keyword, you will be requested to enter the name of the output file. If you press the enter key or enter a space, the output showing the progress

towards the maximum likelihood estimates will appear on the screen and will be lost (note: the final results will be still be available in the file *lung.rwe*). If you want to save this output, enter a name (any name).

(d) Look at the output file with the results *lung.rwe*.

4. Running WEIBULL without covariates

   (a) Open the parameter file *weibull.txt*. Delete the names of the variables after the MODEL statement (to get MODEL;). Comment out the ONLY_STATISTICS statement by adding /* before the statement and */ after the semi-colon. You can also replace ONLY_STATISTICS by STATISTICS. Run WEIBULL. Look at *lung.rwe*.

   (b) Include in the analysis (in *weibull.txt*) the following commands: STD_ERROR; CONVERGENCE_CRITERION (change its value); STORAGE ON_DISK;

   (c) Stratification: Switch back to STORAGE IN_CORE (for better efficiency). Choose a class covariate among the ones available. Sort the recoded data file *lung2.dat* according to this covariate. For example, the *treat* covariate appears in the 4th column (as seen in file *varlist.txt*). To sort the file, you can use Excel:

   Open Excel. From the 'File' menu, click on 'Open'. In the dialog box, choose 'all files' and select *lung2.dat*. Choose the file type ('Delimited') which is not 'Fixed width'. Choose a space as the separator. Select all the file and click on 'Sort' from the 'Data' menu. Choose the columns and order to correctly sort *lung2.dat*.

   Then save the file as a text file (*.txt* or *.prn* with the same name as before). A very frequent error here is that 2 columns stick together into one single column. To avoid this, make sure all your columns are justified to the right before saving *lung2.dat*.

   (d) Include in the analysis the STRATA statement with the name of the covariate you are interested in. Run WEIBULL and look at *lung.rwe*.

5. Running COX without covariates

   (a) Sort the data in file *lung2.dat* in the order required by the COX program (see the User's manual for a detailed explanation). For example, if no stratification is indicated, the first column (time) should be sorted in descending order ; then column 2 (censoring code) should be sorted in ascending order. Call the sorted data file *lung2s.dat*.

   (b) Look at the parameter file *cox.txt*. The only statements not commented out are FILES, TITLE, MODEL and KAPLAN.

(c) Run COX. Look at the results in *lung.rco*.

6. Running WEIBULL with covariates

   (a) In the parameter file *weibull.txt*, include the name of all covariates in the MODEL statement. Comment out the STRATA statement and include in the analysis the TEST statement with the options SEQUENTIAL and LAST. Check that the STD_ERROR statement is included. Run WEIBULL. Look at the results in the file *lung.rwe*. Note that the standard error is not indicated for some of the parameter estimates. Why are those particular effects set to 0? (hint: run WEIBULL again with the" CONSTRAINT LARGEST;" statement and compare the results).

   (b) Run WEIBULL with other CONSTRAINT options (NONE, IMPOSE, FIND). For example, constraint level 2 of the *treat* effect and level 400 of the *cell* effect to 0 (see the CONSTRAINT statement in the User's manual).

   (c) Run WEIBULL with " RHO_FIXED 1;". This corresponds to an exponential regression model. Compare the results and the likelihood value with the Weibull regression model.

   (d) Include as extra continuous covariates the square and the cube of the Karnofsky rating (see the MODEL statement in the User's manual). Should they be included ?

7. Running COX with covariates

   (a) In the parameter file *cox.txt*, include the name of all covariates in the MODEL statement. Comment out the STRATA statement. Sort the data file *lung2s.dat* appropriately.
   Include in the analysis the TEST statement with options SEQUENTIAL and LAST (or some EFFECT options). Check that the STD_ERROR statement is included. Run COX. Look at the results in the file *lung.rco* and compare with the Weibull results for the same model.

   (b) Include BASELINE in the analysis. Try different combinations of the CONSTRAINT statement.

8. Computing predicted records

   (a) Look at the file *lung.fut* which includes 8 individuals. Try to interpret what the characteristics of each one are (e.g., the first individual (number 9101) dies (censoring code = 1) at time 999, after receiving a standard treatment. He is 60 at the beginning of the trial, with a "squamous" type of cancerous cell. His disease was diagnosed 8 months before, he received a prior therapy and his overall status has a value of 60).

(b) Include the FUTURE statement in the parameter file *prepare.txt* and run PREPARE (you can take this opportunity to change the format in "FORMOUT" of *prepare.txt*).

(c) Look at the output file *lung2.fut*.

(d) In the parameter file *weibull.txt*, include the SURVIVAL statement with the options of your choice. Run WEIBULL. Look at the results in the output file *lung.prw*.

(e) Sort the file *lung2.dat* in the appropriate way for the COX analysis. In the parameter file *cox.txt*, include the SURVIVAL statement with the same options as for WEIBULL. Run COX . Look at the results in the output file *lung.prc* and compare with *lung.prw*.

9. Including interactions

(a) Assume that you want to check whether there is an interaction between the *treat* and *cell* effects. To do so, run again the PREPARE program after including the statement :

COMBINE t_by_c = treat + cell;

after the CLASS statement and after adding t_by_c in the OUTPUT list.

(b) Include t_by_c in the WEIBULL and COX analyses. Is the interaction significant ?

(c) Try to find a set of reasonable constraints to express the parameter estimates or their contrast in a sensible form.

2 The Mastitis data set (Gröhn and Hertl, 1996)

2.1 The data

This data set *mast.dat* is a subset of the one analyzed by Gröhn and Hertl in order to study the impact of diseases on culling. It includes records of Holstein cows which calved in two New-York state herds between Jan. 1, 1994 and Dec. 31, 1994 and which were followed until Sept. 30, 1995. The variable of interest is the number of days between the last calving and culling (on a lactation basis). The records of all cows still alive on Sept 30, 1995 or whose lactation under study was followed by another calving were treated as censored. The data were obtained from on-farm software marketed and supported by the Northeast DHIA. The information available for each record consists of :

1. *newid:* the cow number
2. *herdid:* the herd number
3. *mast0:* a variable indicating whether the cow had mastitis (=1) or not (=0) during the lactation
4. *milk60:* the current cumulated milk production during the first 60 days of the lactation (continuous covariate; =0 when not available)
5. *ind60:* an indicator variable equal to 1 when *milk60* is available and equal to 0 otherwise
6. *milk305:* the 305ME milk production during the previous lactation (continuous covariate; =0 when not available)
7. *ind305:* an indicator variable equal to 1 when *milk305* is available and equal to 0 otherwise
8. *days:* the number of days between calving and the culling or censoring date
9. *censor:* the censoring code (=0 for censored cows)
10. *milkcl:* a class of 60-day milk yield (from 1= worst class to 5=best class)
11. *season:* calving season (1=December-February; 2 = March-May; 3 = June-August; 4 = September-November)
12. *prmilkcl* a class of previous 305ME milk yield (from 1= worst class to 5=best class)
13. *parity:* the parity number (from 1 to 6: the 6th class includes parity 6 and above)

14. *mast1:* this is a time-dependent covariate related to mastitis which takes the value 0 as long as the cow is free of mastitis and the value 1 from the day of occurrence of the mastitis until the end of the record.

15. *mast2:* this is a time-dependent covariate related to when the mastitis occurs, which takes the value 0 as long as the cow is free of mastitis and the value 1, 2, 3 or 4 if mastitis occurs between days 0 and 60 (1), 61 and 150 (2), 151 and 270 (3) or after day 270 (4).

This data set illustrates how time-dependent covariates should be included in the COX and WEIBULL programs. In particular, it is essential to have a close look at how the data file *mast.dat* is built. For example, it is important to notice that:

1. There are 15 covariates but, in contrast with the data file in analyses with only time independent covariates, each record consists of at least 16 values: the 16th one contains the *total number of changes in time dependent covariates explicitly indicated in the record*. These changes are presented as sets of consecutive triplets with a standard form (see below). For example, the cow with *newid*=3844 has a value of 0 in column 16: she never had mastitis. The mastitis covariates (in columns 3 , 14 and 15) are always 0. In contrast, the cow with *newid*=3898 has a value of 2 in column 16: there were two changes of time-dependent covariates during the study. This is the consequence of a mastitis occurrence during the lactation (column 3 = 1). These changes are described through the 2 triplets (14,25,1) and (15,25,1).

2. each triplet has the following interpretation:

   (a) The first element is the column number of the covariate whose value changes. e.g., 14 means that the *mast1* variable takes a new value.

   (b) The second element represents the time of the change, e.g., 25 means at day 25.

   (c) The third element contains the new value of the covariate, e.g., 1 means that the new value is 1 (column 14 includes the initial value 0).

   Therefore (15,25,1) means that the covariate *mast2* which was initially equal to 0 (in column 15) switches to a value 1 (mastitis occurring during the period 0-60 days) at time 25.

3. In the particular case studied here, all initial values in columns 14 and 15 for covariates *mast1* and *mast2* are 0 (no mastitis at day 0).

2.2 Analysis

1. Indicating the existence of time dependent covariates
   The parameter file *prepare.txt* in the MASTITIS directory to be used as input the program PREPARE illustrates how the use of time-dependent covariates is specified through the statement:

   TIMEDEP mast1 I4 mast2 I4;

   The I4 simply indicates that the time of change is not expressed as a date but as a number of day since the origin point (see the User's Manual for details).

2. Another type of time dependent covariates
   When the changes in time dependent covariates occur simultaneously for all animals at the same time, there is no need to specify those separately for each record. For example, if we want to study the impact of stage of lactation on culling, we can use the following command:

   TIMECOV stage I4 60 150 270;

   This will generate a new covariate *stage* which takes value 1 before 60 days for all cows and changes to values 2, 3 and 4 at day 60, 150 and 270 for *all* cows. This is equivalent to create a new variable in column n, say with initial value 1 and to include the triplets (n, 60, 2), (n, 150, 3) and (n, 270, 4) for all records.
   Again, I4 indicates that the time change is expressed as a number of days. In other instances, we can have statements like:

   TIMECOV year D6 010190 010191 010192 010193;

   (see the User's manual for details). Note that this new variable *stage* can be used in the COMBINE statement to create interaction variables like *mas1bys = mast1 + stage* , which will be used to study the interaction between *mast1* (mastitis occurrence) and stage of lactation; or *mas2bys = mast2 + stage* , for the interaction between time of occurrence of mastitis *mast2* and stage of lactation.

3. Run PREPARE; check the output files *mast2.dat, codelist.txt, varlist.txt*. Note that for all cows, new "elementary" records have been created, each one corresponding to periods of time when all covariates remained constant. For example, the cow with *newid*= 3846 has three elementary records, one from day 0 to 60 (the code in column 2 is equal to -1, indicating a nonterminal elementary record), one from day 60 to 150 and one from day 150 to 241 (the code in column 2 is equal to 1, indicating death).

4. Run the COX and WEIBULL programs using the parameter files *cox.txt* and *weibull.txt* from the MASTITIS directory as examples. Note that in both files, the analyses are stratified by herd. This has an impact on the way the recoded data file *mast2.dat* should be sorted.

5. Change the model specification in order to check whether there is an interaction between stage of lactation and mastitis and time of occurrence of mastitis.

6. Try to find a meaningful set of constraints which can make easier the interpretation of these interaction terms (hint: constrain the levels of the interaction terms to 0 for the healthy cows, but is it enough?)

7. Fit a *stage* effect with the COX model. How can you interpret the solutions for these effects ?

8. Create a *mast.fut* data set to predict the survivor function of animals with mastitis at different points in time and compare it to the survivor function of healthy cows (note: if you succeed, you are becoming a real expert !)

3 The Seeing Eye data set

The Seeing Eye, Inc. (Morristown, New Jersey) is an educational institution that trains dogs to aid people with reduced or impaired vision. These dogs mainly result from the Seeing Eye breeding program from a closed nucleus of breeding stock. They have been selected for several generations on training ability, suitability of temperament, quality of hips and physical size. Length of time in service is a trait of major interest, economically as well as "emotionally", for the owner of the dog.

3.1 The data

The data set *seye.dat* includes a variety of information regarding the characteristics of each dog:

1. *dogid*: the dog id number.
2. *birthyea*: the birth year of the dog.
3. *birthdat*: the birth date of the dog (as an actual date : DDMMYYYY).
4. *enddate*: the date of failure or censoring (DDMMYYYY).
5. *cens1*: a first type of censoring (for a health related analysis) : if the animal has behavioral problems, its record is considered as censored at the end of service.
6. *cens2*: a second type of censoring (for a more general analysis) : only animals still alive at the end of the study period (01051996) are considered as censored.
7. *breed*: the breed of the dog (1= Labrador; 2 = German Shepherd).
8. *sex*: the sex of the dog (1 = male; 2 = female).
9. *sire*: the id number of the sire of the dog.
10. *dam*: the id number of the dam of the dog.
11. *inbred*: the inbreeding coefficient of the dog in percent.
12. *gener*: the "equivalent" generation number since the base generation.
13. *xray_wt*: the weight of the dog when the dog is X-rayed for the evaluation of hip quality (in pounds).
14. *cgrp_hip*: a number identifying all animals of a same breed X-rayed the same calendar quarter

15. *hipscore*: a score for the quality of the hip of the dog on a scale from 1 to 9. Hip scores of 1, 2 or 3 are considered as dysplastic. Dysplastic animals are NOT entering service.

16. *xr_dev*: the weight of the dog *xray_wt* expressed as deviation from the contemporary group mean.

17. *hips_dev*: the hip score *hipscore* expressed as deviation from the contemporary group mean.

18. *cgrp_tem*: a number identifying all animals of a same breed scored for temperament on the same calendar quarter.

19. *temper*: an objective measure of temperament and overall training attitude ("first blindfold test") on a scale from 1 to 9. Animals with scores below 4 are NOT entering service.

20. *temp_dev*: temperament measure expressed as deviation from the contemporary group mean.

21. *badhips*: an indicator variable equal to 1 if the hip score of the dog is one phenotypic standard deviation below the contemporary group average.

A second data set *seye.ped* includes the pedigree information. This is in fact a repeat of the pedigree information in *seye.dat*. The variables *sex, sire* and *dam* could have been deleted from *seye.data*

1. *dogid*: the dog id number.

2. *sex*: the sex of the dog (1 = male; 2 = female).

3. *sire*: the id number of the sire of the dog.

4. *dam*: the id number of the dam of the dog.

In this data set, one can see how to handle dates and how to incorporate a random effect with a relationship matrix in the survival analysis. It also illustrates some of the problems in the computation of the parameter estimates (here, with the WEIBULL model) and interpretation of the results.

3.2 Analysis

1. Look at the parameter file *prepare.txt* in the SEYE directory. Notice how dates are treated in the INPUT, TIME, TIMECOV statements (for the creation of a new time dependent covariate *year* for the latter). See also how the pedigree information (*seye.ped* file) is recoded.

2. Run PREPARE (Reminder: You may have to change some paramters related to the database size. If so, do this via the appropriate keywords, e.g. NRECMAX.)

3. Look at the parameter file *weibull.txt* and modify it in order to get statistics (only) on the variables that you consider having a potential impact on length of service. Run WEIBULL and check the results in the file *seye.rwe*.

4. Sort the recoded data file for the COX program (with or without strata, as you want).

5. Run the COX program until you find a fixed effect model that seems reasonable. Interpret the results: which effect influences time in service?

6. Add the *sire* effect as a random effect. First assume a normal distribution with variance 0.02. Run COX.

7. Compare the *sire* effect estimates with a COX model where the *sire* is assumed to have a log-gamma distribution with gamma parameter $\gamma = 50 = 1/0.02$. What do you conclude?

8. Add the ESTIMATE statement and change the value 0.02 into a group of three values (see the User's Manual). The values needed are of the type "*lower bound, upper bound, final tolerance*", e.g., "0.005 0.1 0.001" to estimate the sire and dam variance (assumed to be equal here). Observe the bisection process used to find the mode of the marginal posterior distribution of the variance parameter (indicated as BEST VALUE FOUND).

9. Start again after adding the MOMENTS statement, in order to also get the mean, the standard deviation and the coefficient of skewness of that same marginal posterior distribution.

10. Sort the recoded data file *seye2.dat* for the WEIBULL program. Run WEIBULL with the same fixed model as for COX. What happens ?

11. Add the RHO_FIXED statement choosing an appropriate value for ρ. Proceed as for the previous COX analysis ... until you are stuck...

4 The (simulated) animal breeding data set

4.1 The program

The program *simul.f* generates N (possibly censored) failure time records from a Weibull distribution with user specified characteristics. The generation of these records can be influenced by fixed effects (up to 3) and a (normal) random effect (optional) that we will call a *sire* effect. Each of the N_S sires has N/N_S daughters (balanced design). A simple relationship matrix between sires (through "sires of sires") can also be generated. The source code and the Windows executable is supplied to the Survial Kit. On a UNIX platform, it should be compiled as:
xlf -osimul simul.f
and run as: simul

4.2 Simulated data

When running the program, you are asked to enter sequentially:

1. a "seed" (any integer number) to be used in the random number generation process;
2. the number of records to simulate;
3. the Weibull parameter ρ and the median value of failure time (which defines the other Weibull parameter ρ);
4. the number of fixed effects you want to generate (1 to 3);
5. for each fixed effect: the number of levels and an interval (2 values) for the relative risk associated with this effect;
6. whether you want to include a random effect;
7. if the answer is yes:
   (a) the number of sires;
   (b) the variance of the random effect (a normal distribution is always assumed);
   (c) whether you want or not to include relationships between sires and if yes, the number of sires of sires (balanced design again);

8. a censoring time (same for all records). If the value entered is 0 or a very large value, all records will be uncensored. However, for later use in COX and WEIBULL (for which a largest value of failure time is specified in the *parinclu* file (parameter NTIMMAX)), it is better to include a reasonable upper value in all cases.

The output files are *datsim*, the simulated data set; *pedigs*, the pedigree file for the sires (if relevant) and *true*, a file containing the true solutions for fixed and random effects. The variables in the data file *datsim* are:

1. an id number
2. the simulated failure time
3. the censoring code (0 = censored; 1 = uncensored)
4. the level of each fixed effect (up to 3 values)
5. the sire id number if a random effect is specified

The program *simul.f* offers an (admittedly simplistic) check at the way the distribution parameters of a random effect can be estimated using the Survival Kit.

4.3 Analysis

1. Run *simul* entering the parameters of your choice.
2. Create your own input parameter file for PREPARE (hint: it is easier if you copy and then modify one of the parameter files already used).
3. Run PREPARE. Sort the recoded data file as required. Create your own COX and WEIBULL parameter files. Run COX and WEIBULL. Compare true and estimated solutions. Compare the COX and WEIBULL solutions, too.
4. If a random effect was generated, estimate its variance with the COX program. Do the same thing with the WEIBULL program. Compare the results between the two models and the true variance.
5. Rerun COX and WEIBULL and estimate the gamma parameter assuming a log-gamma distribution for the *sire* effect. Relate it to the true variance of the normal distribution.

6. With the WEIBULL program (and only with WEIBULL), there is another way to estimate the fixed effects and the gamma parameter:

   Sort the recoded data set by increasing *sire* number. This sort is necessary if you want to integrate out the sire effect (see the User's manual), in order to obtain the marginal posterior distribution of the fixed effects, the Weibull parameters and the gamma parameter. Include:
   RANDOM sire loggamma x;
   INTEGRATE_OUT sire;
   where x is the inverse of the true normal variance. Then run WEIBULL again. Compare the results (fixed effects, tests) with the previous ones.

7. Estimate the gamma parameter γ for the *sire* effect. Two alternative ways are possible:

   (a) You can estimate γ jointly with the other fixed effects but after algebraic integration of the *sire* effect:
   RANDOM sire loggamma *any_x*;
   INTEGRATE_OUT JOINT_MODE sire;
   Here, *any_x* is used as a starting value.

   (b) or you can estimate γ after (Laplacian) integration of all the other parameters from its marginal posterior distribution:
   RANDOM sire estimate moments loggamma *lower_bound upper_bound tolerance*;
   INTEGRATE_OUT sire;