IMPUTING IMPLANT DATES

JAN DE LEEUW AND CHRISTINE PENG

CONTENTS

1. Problem 1
2. Research Data Base 1
2.1. Data Base Size 1
2.2. Lags for Complete Records 2
3. Brookmeyer Procedure 2
3.1. Homogeneity 2
3.2. Ignorability 3
3.3. Use of the Normal Disturbance 3
3.4. Use of a Single Imputation 4
Appendix A. Lag Statistics 6
Appendix B. Return Rates 7
Appendix C. Regression Analyses for Lag 8
Appendix D. C Programs 10
Appendix E. Lisp Programs 13

1. PROBLEM

The Research Data Base (RDB, from now on) has information on more than 85,000 valves. For all of these valves we know invoice dates, but for less than half of them we know implant dates. The problem we investigate in these notes is if implant dates can be imputed reliably from the available information. The investigation will also have implications for the imputations performed by Shiley, which are at the basis of the distribution of implants over the 8 Oct-Sep periods. This distribution defines the at-risk population for the survival analysis, and is consequently vital for subsequent calculations.

For our information on what Shiley has done with the RDB, we have at our disposal the deposition of October 14, 1995, by Dr. Steven Lewis, plus several exhibits going with that disposition. In particular, the disposition refers to the procedures followed by Dr. Brookmeyer,

Date: November 5, 1996.

explained in Exhibit 8, section III-A, page 20, or in Exhibit 18, Appendix A.

All computations in these notes have been double-checked. They were independently programmed by Jan de Leeuw in the C language, and by Christine Peng in the SAS system.

2. Research Data Base

2.1. Data Base Size. The RDB has 85,756 records. We selected those records for which an implant card was available. There are 36,363 of these. In addition, there are 17 cases with an implant card, but with either invoice date or implant date missing. These were eliminated as well, which leaves 36,346 records. If we eliminate all records for which either the invoice date or the implant date is missing, independently of the implant card, we have 38,369 records.

2.2. Lags for Complete Records. The mean lag-time for US is 181 days, for non-US it is 353 days. This is very different from the values reported in the Lewis disposition and the Brookmeyer exhibits, which give 94 and 184 days, respectively. We do not have an explanation for these enormous differences. It is unlikely that the Brookmeyer estimates are different because they include imputed values. As the tables in the appendix show, imputation means adding a large number of non-US implant times, i.e. add a large number of lag-times above the average.

Another somewhat surprising fact is that there are also 110 negative lag-times in the RDB, by the way, for which the implant seems to happen before the invoice.

3. Brookmeyer Procedure

The RDB has a variable `EST_IMPL`, which contains the estimated implant date for all valves. From the information provided to us it is not possible to reconstruct precisely how these estimated implant dates were computed.

It is clear from Exhibit 8, and from the disposition, that regression analysis was used, but it is not entirely clear which predictors were used in the Brookmeyer construction of `EST_IMPL`. The discussion suggests that the outcome variable chose was the log of the lag-time in days, with as predictors US versus non-US (i.e. categories 01-09 of `GEOCODE` vs categories 10-29), and the invoice date. But we do not know if the logarithm of invoice date was used, or a binary variable indicating whether the invoice was in the first three years or the second three years. In any case, in Appendix C are the results of a regression analysis which uses log-lag as outcome, and US/non-US and invoice-date as predictors.After the regression analysis implant times are imputed by using the regression formula to predict the log-lag. Then a normal deviate with mean zero and standard deviation (Root MSE from the analysis) 1.21 is added. Then the lag is computed by taking the exponent, and the lag is added to the invoice time to get the estimated implant time.

The fact that R^2 is only 8% indicates that lag (and thus implant time) cannot be predicted very well from these two predictors, and thus there will be a great deal of uncertainty in the imputed implant times. For completeness we also give the results of a regression analysis which use invoice-date and GEOCODE. The R^2 increases to 13%

3.1. **Homogeneity.** In imputing implant times, the Brookmeyer procedure only distinguishes US and non-US implants. But GEOCODE has 29 different values, of which nine describe US groups (eight hospitals and one rest category), and 20 describe non-US groups (19 countries and one rest category). Some pertinent statistics comparing the 29 groups are given in Table 1 in the Appendix. It is clear that the differences are dramatic.

A simple analysis of variance show that the within-domestic variance is highly significant ($F = 71.42$ with 8 and $25,467$ dfr, $p < .0001$), i.e. the nine hospitals in the US have very different lag distributions. For the 20 countries, we have $F = 67.04$ with 19 and $10,850$ dfr, $p < .0001$. The countries also have very different lag distributions.

3.2. **Ignorability.** Imputation procedure almost invariably rely on the assumption of *ignorability.* This means we assume that the distribution of lags in the population for which we have the implant date is the same as the distribution for which we do not have the implant date.

Obviously, this assumption cannot be tested directly, because we do not know the lag distribution if we don't have the implant date. But the reasoning behind the assumption is that implant dates are *missing at random*, and we expect all variables in the data set to have the same distribution for IMPCARD is "Y" and IMPCARD is "N".

We do not investigate ignorability for all variables, but we just look at GEOCODE, because we have seen that certainly GEOCODE is related to lag-time. If certain hospitals or countries send back implant carts at different rates, then the lag-distributions in the two populations will tend to be different.

The results are in Table 2 in the Appendix. Obviously return rates are wildly different for different geocodes. There seems to be a clear tendency for countries with longer lag times to return fewer implant cards. This means that lag times will tend to be underestimated by imputation.

3.3. **Use of the Normal Disturbance.** The Brookmeyer procedure uses the normal distribution to implement imputation uncertainty. Thus a normal variate with mean zero and variance equal to the residual variance around the regression line is added to the estimated implant date. This raises the question how appropriate the normal distribution really is in this case. Remember that we want distributions of actual and imputed lag times to be the same. If the distribution of the actual lag times is different from a normal one, then the distribution of the imputed lag times should not be normal either. We can distinguish the assumption of normality for the lags and for the logarithms of the lags. Both were tested using the Shapiro-Wilks test. For lag we cannot reject the hypothesis of normality for Jordan and Greece (with only 3 and 6 cases, respectively). If we take log-lag, then normality cannot be rejected for Allegheny, Iowa, Mayo, Alabama, Switzerland, Jordan, and Hong Kong. In the other 22 geocodes, the hypothesis of normality must be rejected.

For imputation purposes, making the strong and unrealistic assumption of normality is not at all necessary. A simple nonparametric imputation procedure is as follows. Suppose we have n known lags for a particular geocode. To impute an implant date in that geocode we draw one of the n lags at random and add it to the invoice date. For the next imputed implant date we draw another lag, with replacement, from the n known ones. And so on, until all implant dates are imputed.

3.4. **Use of a Single Imputation.** Statistical procedures are incomplete, and potentially misleading, if they do not give information about the variability (the standard error or confidence interval) of their outcomes. In the imputation of implant times, there are two basic sources of uncertainty.

The first one has to do with the inaccuracy in the RDB. The RDB may not cover all relevant valves, some valves in the RDB may not have been implanted, the RDB may contain coding errors, and so on. There is no way we can get a handle on the amount of distortion introduces by these factors, but clearly some of it is there. In the RDB, for example, there are these 110 records (of the $36,346$) for which the implant date precedes the invoice date.

The second source of uncertainty is uncertainty due to imputation. If we repeat our imputation procedure, we will get different imputed values, because of the random elements involved (either adding a normally distributed disturbance, or taking a random sample with replacement of the known lags). To get an idea of the variability, we have to repeat our imputation procedure a large number of times, and note the differences in the results. We need *multiple imputation.* The Brookmeyer reports, and the Lewis implant distributions, do not give information about this variability, and are consequently incomplete and possibly misleading.

This can be illustrated quite simply with an example. Suppose we use the time-intervals in Exhibits 5 and 6, and we impute using the appropriate normal distribution for each geocode seperately. For geocode 01 (Deborah Heart and Lung Institute) we have a mean lag of 68 and a standard deviation of 139. Let us select a valve invoiced at 12/01/81, for which we want to impute the implant. We add a normal deviate with mean 68 and standard deviation 139, and we find for the eight intervals the following imputation probabilities.

–79	79–80	80–81	81–82	82–83	83–84	84–85	85–
.0000	.0001	0.1765	0.7785	0.0448	0.0000	0.0000	0.0000

But for geocode 10, The Netherlands, the probabilities are (for the same date),

–79	79–80	80–81	81–82	82–83	83–84	84–85	85–
.0093	.0374	0.1106	0.2121	0.2632	0.2118	0.1096	0.0460

Thus we see that for The Netherlands imputation in three of the yearly intervals is about equally likely, while the additional two neighboring intervals are quite probable as well. Since for the Netherlands about 80% is imputed, this means the implant dates are quite arbitrary. They are much better determined for the US sites, but there is not much imputation going on in these sites anyway. The Netherlands is also a rather favorable case, because of the high number of implants and the relatively high return rate.

Appendix A. Lag Statistics

GEOCODE	N	Mean	Median	StaDev
Deborah Heart and Lung Institute	1284	67.99	43.0	139.245
Alleghany Hospital	93	181.16	53.0	432.560
St. Lukes Hospital	581	124.71	91.0	119.028
Iowa	530	259.37	162.0	297.536
Lankenau Hospital	563	106.42	67.0	159.800
Thomas Jefferson Hospital	545	129.72	90.0	141.352
Mayo Clinic	521	127.07	85.0	135.918
University of Alabama	510	187.98	118.0	208.389
USA, Other	20849	192.33	111.0	239.461
Netherlands	1779	484.78	324.0	543.062
France	684	530.84	196.5	768.268
U.K.	1965	282.59	228.0	218.705
Germany	413	529.33	265.0	682.722
Sweden	517	584.49	188.0	764.136
Spain	701	291.85	166.0	447.173
Japan	340	247.66	196.0	204.522
India	39	400.49	258.0	625.299
Canada	1858	172.47	115.0	194.780
Italy	269	438.70	226.0	567.220
Belgium	67	659.60	257.0	846.722
Switzerland	263	391.58	204.0	539.762
Iraq	8	1930.38	2269.0	821.109
Uruguay	666	191.64	128.5	222.538
Australia	437	170.32	105.0	254.245
Jordan	3	318.00	21.0	806.608
Greece	6	2130.00	2337.0	604.274
Hong Kong	20	452.80	322.5	328.430
Mexico	1	412.00	412.0	.
Other International	834	488.02	316.5	513.454

Table 1. Lag Statistics for each GEOCODE

APPENDIX B. RETURN RATES

GEOCODE	N	Return Rate
Deborah Heart and Lung Institute	1403	91.52
Alleghany Hospital	211	44.08
St. Lukes Hospital	599	96.99
Iowa	605	87.60
Lankenau Hospital	574	98.26
Thomas Jefferson Hospital	579	94.13
Mayo Clinic	572	91.08
University of Alabama	557	91.56
USA, Other	26134	79.81
Netherlands	8832	20.14
France	7978	8.57
U.K.	5385	36.51
Germany	4482	9.21
Sweden	3836	13.48
Spain	3489	20.09
Japan	2814	12.08
India	1956	1.99
Canada	1977	94.23
Italy	1726	15.59
Belgium	1180	5.68
Switzerland	1128	23.32
Iraq	846	0.95
Uruguay	904	73.78
Australia	894	48.88
Jordan	835	0.36
Greece	713	0.84
Hong Kong	652	3.07
Mexico	599	0.17
Other International	4296	19.41

TABLE 2. Implant Card Return Rate per GEOCODE

Appendix C. Regression Analyses for Lag

```
                        Analysis of Variance

                             Sum of         Mean
Source            DF        Squares       Square      F Value

Model              2     4548.53408   2274.26704     1554.826
Error          36343    53159.43527      1.46271
C Total        36345    57707.96934

      Root MSE       1.20943     R-square       0.0788
      Dep Mean       4.76531     Adj R-sq       0.0788
      C.V.          25.37981

                     Parameter Estimates

                  Parameter      Standard    T for H0:
Variable  DF       Estimate         Error   Parameter=0

INTERCEP   1       5.554026    0.02748348       202.086
US         1      -0.743119    0.01394542       -53.288
INVOICE    1      -0.000116    0.00001132       -10.293
===================================================================
                       Analysis of Variance

                                   Sum of         Mean
Source            DF        Squares       Square      F Value

Model             29     7503.70722    258.74852      187.170
Error          36316    50204.26213      1.38243
C Total        36345    57707.96934

      Root MSE       1.17577     R-square       0.1300
      Dep Mean       4.76531     Adj R-sq       0.1293
      C.V.          24.67345

                               Parameter Estimates

                       Parameter      Standard     T for H0:
Variable  DF       Estimate         Error   Parameter=0    Prob > |T|

INTERCEP   1       6.020768    0.04808198       125.219        0.0001
INVOICE    1      -0.000145    0.00001169       -12.423        0.0001
PLACE1     1      -2.002604    0.05234596       -38.257        0.0001
```

```
PLACE2      1     -1.614351    0.12854408        -12.559         0.0001
PLACE3      1     -1.219364    0.06363275        -19.163         0.0001
PLACE4      1     -0.754531    0.06561429        -11.499         0.0001
PLACE5      1     -1.640700    0.06413830        -25.581         0.0001
PLACE6      1     -1.332124    0.06476459        -20.569         0.0001
PLACE7      1     -1.267504    0.06566606        -19.302         0.0001
PLACE8      1     -1.004735    0.06631318        -15.151         0.0001
PLACE9      1     -1.077116    0.04159825        -25.893         0.0001
PLACE10     1      0.060511    0.04936911          1.226         0.2203
PLACE11     1     -0.237121    0.06068037         -3.908         0.0001
PLACE12     1     -0.330023    0.04861345         -6.789         0.0001
PLACE13     1     -0.035302    0.07079633         -0.499         0.6180
PLACE14     1     -0.266526    0.06608119         -4.033         0.0001
PLACE15     1     -0.583833    0.06029424         -9.683         0.0001
PLACE16     1     -0.468322    0.07565616         -6.190         0.0001
PLACE17     1     -0.131463    0.19307491         -0.681         0.4959
PLACE18     1     -0.951276    0.04900664        -19.411         0.0001
PLACE19     1     -0.220463    0.08244369         -2.674         0.0075
PLACE20     1     -0.048552    0.14930311         -0.325         0.7450
PLACE21     1     -0.408426    0.08327226         -4.905         0.0001
PLACE22     1      1.420081    0.41775318          3.399         0.0007
PLACE23     1     -0.817809    0.06133605        -13.333         0.0001
PLACE24     1     -1.138150    0.06943651        -16.391         0.0001
PLACE25     1     -2.307384    0.68004963         -3.393         0.0007
PLACE26     1      1.792303    0.48182948          3.720         0.0002
PLACE27     1      0.345892    0.26629740          1.299         0.1940
PLACE28     1      0.357190    1.17647597          0.304         0.7614
```

Appendix D. C Programs

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define STARTYEAR 76
#define DEBUG 0

int day_of_year (int year, int month, int day);
int days_since (int year, int month, int day);
int is_leap (int year);

char daytab[2][13] =
{
  {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
  {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
};

main ()
{
  FILE *infile = fopen ("ccvalves.dat", "r"), *outfile = fopen ("dates.out", "w"),
   *fracfile = fopen ("fracs.out", "w"), *holland = fopen ("holland.out", "w");

  char INVCDATE[20], SERIAL[20], VTYPE[20], AGE[20], SEX[20], IMPLPOS[20],
    IMPLDATE[20], IMPCARD[20], GEOCODE[20], SHILDATE[20], WELDDATE[20],
    EXPLDATE[20], FRACTURE[20], PRODDES[20], SERIALNO[20], DISCFIT[20],
    DHRLDATE[20], OANGLE[20], SHRINK[20], LOADDEFL[20], DISCSTUT[20], SHOPORD[20],
    EVENTDT[20], REMILL[20], REWORK[20], OUTCOME[20], VSTATUS[20], DOB[20],
    EST_IMPL[20], RISKRATE[20], WELDER1[20], WELDER2[20], WELDER3[20],
    WELDER4[20], FLANGE[20], SEW_RING[20];

  char record[245], RESID[2];

  int invcyear, invcmonth, invcday, implyear, implmonth, implday, evntyear,
    evntmonth, evntday, lag;

  int i = 0;

  while (fgets (record, 245, infile))
    {

    sscanf (record,
"%8c%18c%12c%12c%1c%10c%8c%1c%2c%8c%8c%8c%1c%10c%6c%8c%8c%2c%1c%1c%3c%10c%8c%1c%12
INVCDATE, SERIAL, VTYPE, AGE, SEX, IMPLPOS, IMPLDATE, IMPCARD, GEOCODE, SHILDATE,
WELDDATE, EXPLDATE, FRACTURE, PRODDES, SERIALNO, DISCFIT, DHRLDATE, OANGLE, SHRINK
DISCSTUT, SHOPORD, EVENTDT, REMILL, REWORK, OUTCOME, VSTATUS, DOB, EST_IMPL, RISKR
WELDER1, WELDER2, WELDER3, WELDER4, FLANGE, SEW_RING, RESID);

    INVCDATE[8] = SERIAL[18] = VTYPE[12] = AGE[12] = SEX[1] =
IMPLPOS[10] = IMPLDATE[8] = IMPCARD[1] = GEOCODE[2] = SHILDATE[8] =
WELDDATE[8] = EXPLDATE[8] = FRACTURE[1] = PRODDES[10] = SERIALNO[6] =
DISCFIT[8] = DHRLDATE[8] = OANGLE[2] = SHRINK[1] = LOADDEFL[1] =
DISCSTUT[3] = SHOPORD[10] = EVENTDT[8] = REMILL[1] = REWORK[12] =
OUTCOME[1] = VSTATUS[3] = DOB[8] = EST_IMPL[12] = RISKRATE[12] =
WELDER1[4] = WELDER2[4] = WELDER3[4] = WELDER4[4] = FLANGE[12] =
SEW_RING[12] = '\0';

    if ((GEOCODE[0] == '1') && (GEOCODE[1] == '0'))
fprintf (holland, "%s %s %s %s %s %s\n", INVCDATE, IMPLDATE, IMPCARD,
FRACTURE, EVENTDT, EST_IMPL);

    if ((IMPLDATE[0] != ' ') && (INVCDATE[0] != ' '))

{
sscanf (INVCDATE, "%d/%d/%d", &invcmonth, &invcday, &invcyear);
sscanf (IMPLDATE, "%d/%d/%d", &implmonth, &implday, &implyear);

  invcday = days_since (invcyear, invcmonth, invcday);
  implday = days_since (implyear, implmonth, implday);
      lag = implday - invcday;
  fprintf (outfile, "%5d %5d %5d %s %s\n", invcday, implday, lag, GEOCODE, EST_IMP

  if ((FRACTURE[0] == 'Y') && (EVENTDT[0] != ' '))
  {
      sscanf (EVENTDT, "%d/%d/%d", &evntmonth, &evntday, &evntyear);
      evntday = days_since (evntyear, evntmonth, evntday);
      lag = evntday - implday;
      fprintf (fracfile, "%5d %5d %5d %s\n", implday, evntday, lag, GEOCODE);
  }

}
if(DEBUG)
   {
   if (i == 500)
   exit(0);
   i++;
   }
   }
  fclose (outfile);
  fclose (infile);
  fclose (fracfile);
  fclose (holland);
}

int
day_of_year (int year, int month, int day)
{
  int i, leap;

  leap = is_leap (year);
  for (i = 1; i < month; i++)
    day += daytab[leap][i];
  return day;
}

int
is_leap (int year)
{
return year % 4 == 0 && year % 100 != 0 || year % 400 == 0;
}

int
days_since (int year, int month, int day)
{
int theday = day_of_year (year, month, day), i;

for (i = STARTYEAR + 1; i <= year; i++)
theday += 365 + is_leap(i);
return theday;
}
```

Appendix E. Lisp Programs

```
(defun date-converter (year month day)
  (let ((daytab (make-array '(2 13) :initial-contents
               '((0 31 28 31 30 31 30 31 31 30 31 30 31)
         (0 31 29 31 30 31 30 31 31 30 31 30 31))))
        (leap (if (leap year) 1 0)))
    (dotimes (i month day)
        (setf day (+ day (aref daytab leap i))))
))

(defun days-since (year month day &optional (start 0))
(let ((theday (date-converter year month day))
      (thegap (- year start)))
  (dotimes (i thegap theday)
    (setf theday (+ theday (if (leap i) 366 365))))
))

(defun leap (year)
  (and (= 0 (mod year 4))
       (or (not (= 0 (mod year 100)))
       (= 0 (mod year 400))))
)

(defun interval (x mean stdev)
  (let* ((bnds '(1370 1736 2101 2466 2831 3197 3562))
         (cdfs (normal-cdf (/ (- bnds (+ x mean)) stdev))))
    (difference (concatenate 'list '(0) cdfs '(1)))
))

#|
> (days-since 85 3 1)
31107
> (days-since 1985 3 1)
725067
> (days-since 1985 3 1 1976)
3348
|#
```

UCLA Statistical Consulting Center, 8118 Math Sciences Building,
405 Hilgard Ave, Los Angeles, CA 90095
E-mail address: consult@stat.ucla.edu