Homework # 4: R Code

Scott S. Emerson, M.D., Ph.D.

Biost 514/517

October 25, 2012

Abstract

The following code illustrates the R commands that could be of use to answer Homework # 4.

1 Initializing Functions and Datasets

We first source the descriptive R functions (and load the survival library) by typing:

```
> source("http://www.emersonstatistics.com/courses/formal/b517_2012/uDescriptives.txt")
> library(survival)
```

We then read in the salary data as a `"data.frame"` object named `SALARY`. Note that I read in the character strings as character strings rather than factors

```
> SALARY <- read.table("http://www.emersonstatistics.com/datasets/salary.txt",
+         header=TRUE,        stringsAsFactors=FALSE)
```

The variables in the resulting data frame are:

```
> names(SALARY)
```

```
 [1] "case"    "id"      "sex"     "deg"     "yrdeg"   "field"   "startyr"
 [8] "year"    "rank"    "admin"   "salary"
```

2 Creating the Variable Measuring Time as Associate Professor

We want to identify all subjects for whom the year they became an associate professor is known. For this to be true

- They must have been hired by the university as an assistant professor.
- They must have been promoted to associate professor after 1976.
- They must still be working for the university in 1995.

We thus try to find the earliest year that each faculty member was an associate professor. The following code will do this (subjects who were never associate professor at the university between 1976 and 1995 will have the missing data code `NA` for this variable. I choose to include the new variable in the `SALARY` data frame.

```
> SALARY$fstAssoc <- clusterStats(SALARY$year, cluster=SALARY$id, stat="min",
+         subset=SALARY$rank=="Assoc")
```

We do not know when a faculty member was promoted to associate professor if they were never an associate professor between the years 1976 and 1995. As noted above, these subjects will already have a `NA` value for `fstAssoc`.

We do not know when a faculty member was promoted to associate professor if they were an associate professor in 1976. (Our data only goes back to 1976). These people will have `fstAssoc` equal to 76.

We also do not know when a faculty member was promoted to associate professor if they were hired as an associate professor. These people will have `fstAssoc` equal to their value for `starter`.

We will have to exclude the people in the last two categories from our analysis by setting `fstAssoc` to `NA`. We can do that with the following code, which makes use of the function `ifelse()`.

```
> SALARY$fstAssoc <- ifelse( SALARY$fstAssoc==76 | SALARY$fstAssoc==SALARY$startyr,
+         NA,  SALARY$fstAssoc)
> descrip(SALARY$fstAssoc)

                     N     Msng  Mean      Std Dev   Geom Mn    Min
SALARY$fstAssoc:   19792 11235   84.72     5.119     84.57     77.00
                    25%       Mdn       75%       Max
SALARY$fstAssoc:     80.00     84.00     89.00     95.00
```

Now we do analogous commands to find the first year a faculty member was a full professor in our data set.

```
> SALARY$fstFull <- clusterStats(SALARY$year, cluster=SALARY$id, stat="min",
+         subset=SALARY$rank=="Full")
> descrip(SALARY$fstFull)

                    N     Msng  Mean      Std Dev   Geom Mn    Min       25%
SALARY$fstFull:   19792  6278   83.66     6.325     83.42     76.00     77.00
                     Mdn       75%       Max
SALARY$fstFull:     83.00     89.00     95.00
```

Note that `fstFull` might be equal to 76 for some faculty members and other might have `fstFull` equal to `starter`. But all of those people with have `fstAssoc` equal to `NA`.

Note that `fstFull` will be `NA` for anyone who was always an assistant professor prior to 1996, but that these people will have `fstAssoc` equal to `NA`.

Note that `fstFull` will be `NA` for anyone who was an associate professor in 1995. If these people were hired as an assistant professor and promoted after 1976, these people will have a non missing value for `fstAssoc`.

Now we can compute the time a faculty member in our data spent as an associate professor after we have observed him/her having been promoted to assistant professor. We must address the possibility that

a faculty member is still an associate professor in 1995. I do this by using the value in `fstFull` if it is non missing, and using 95 otherwise. I use the function `ifelse()` with the condition `is.na(SALARY$fstFull)`, where the function `is.na()` returns `TRUE` if its argument has a missing value, and it return `FALSE` if not.

```
> SALARY$ttofull <- ifelse(is.na(SALARY$fstFull),  95, SALARY$fstFull) - SALARY$fstAssoc
> descrip(SALARY$ttofull)

                    N     Msng  Mean      Std Dev   Geom Mn    Min       25%
SALARY$ttofull:   19792 11235   6.824     3.987      NA       0.0000    4.000
                   Mdn       75%       Max
SALARY$ttofull:     6.000     9.000     18.00
```

Notice that I did not need to worry about cases where the faculty member was never observed to be promoted to associate professor, because all those subjects have `NA` for `fstAssoc`, and thus will have `NA` for `ttofull`.

But `ttofull` now contains some right censored data. So we have to create an indicator of who was promoted.

We can use the above information to decide who we saw both promoted to associate and later promoted to full. I use a logical statement to create a true/false value, but then I convert it to an integer (in which case `TRUE` becomes 1, and `FALSE` becomes 0, and `NA` stays `NA`).

```
> SALARY$promoted <- as.integer( !is.na(SALARY$fstAssoc) & !is.na(SALARY$fstFull) )
```

Note that after creating the above code, every single subject has a non missing value for `promoted`. But the faculty members we are uninterested in will have a missing value for `ttofull`, so when I perform Kaplan-Meier analyses those subjects will be ignored. If this bothers you, however, we can replace the values with `NA`:

```
> SALARY$promoted[ is.na(SALARY$ttofull) ] <- NA
```

2.1 `Surv` Objects in R

Our variables `ttofull` and `promoted` can not be interpreted in a scientifically meaningful way by themselves. Together they represent right censored data. We indicate that to R by creating a `"Surv"` object. As I have no particular reason to keep a separate variable for the possibly censored times in `ttofull`, I just re-assign the variable `ttofull` to represent my `"Surv"` object.

```
> SALARY$ttofull <- Surv( SALARY$ttofull, SALARY$promoted )
```

All that a `"Surv"` object is in this case is a two column matrix, where the first column is the potentially right censored observation times, and the second column is the indicator variable denoting which observations represent an event. But that matrix is now labeled with the `"Surv"` class, and R knows to analyze and print that variable using special techniques. When the values in a `"Surv"` object are printed, censored observation have a "+" sign appended to the value.

3 Computing Kaplan-Meier Estimates of the Survival Function

We obtain Kaplan-Meier estimates of a survival function by using `survfit()` to create a `"survfit"` object containing the estimates and other pertinent information. The `survfit()` function takes a formula where

you indicate any strata you want to use. If you want the Kaplan-Meier estimates for the entire sample, you indicate this by using '1' in the formula:

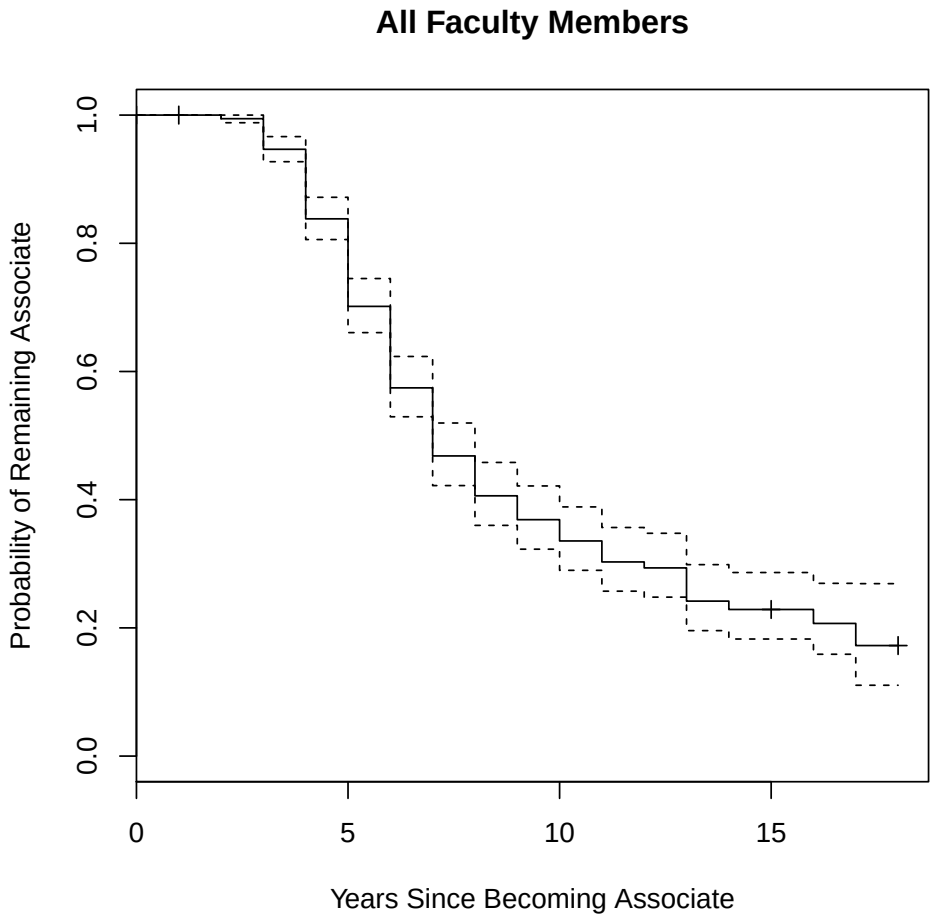
```
> allKM <- survfit (SALARY$ttofull ~ 1, subset= SALARY$year==95)
> sexKM <- survfit (SALARY$ttofull ~ factor(SALARY$sex),
+         subset= SALARY$year==95)
```

Note that in the above code, I used a subset defined by the data from year 1995. The value of `ttofull` was identical for every row in the data set that pertained to the same faculty member, but I should only represent each faculty member once in my data analyses. Our sampling scheme included only those subjects who were employed at the university in 1995, so I know that each faculty member has such a record. Thus that becomes my "tag" for ensuring only one observation is used for each individual.

From the above code, I now have a `"survfit"` object named `allKM` for the entire sample, and one named `sexKM` that contains stratified survival curve estimates. Note that when using a character string variable in the formula, I have to identify it as a "factor".

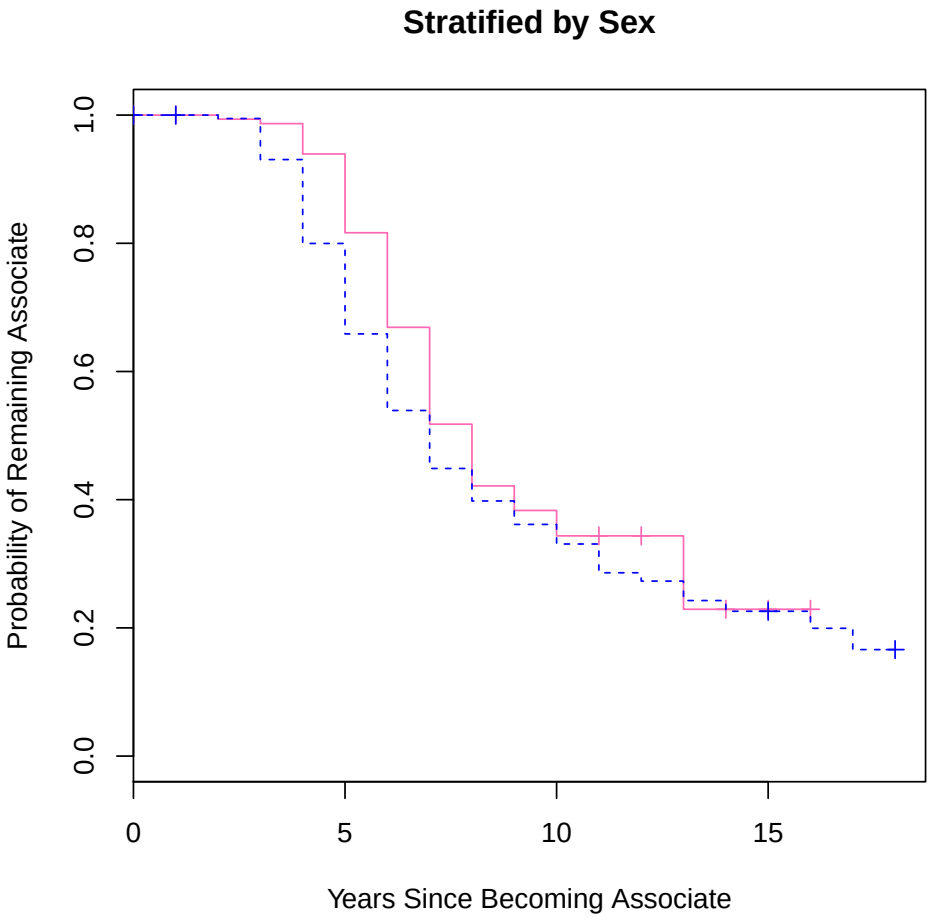
You now can produce plots of the survival curve (and CI by default) for the entire sample using code like

```
> plot(allKM, main="All Faculty Members",         xlab= "Years Since Becoming Associate",
+         ylab="Probability of Remaining Associate")
```



You now can produce plots of the survival curve stratified by sex using code like (note that the strata will use colors and line types in alphabetical or numeric order of the strata names).

```
> plot(sexKM, main="Stratified by Sex",
+         xlab= "Years Since Becoming Associate",
+         ylab="Probability of Remaining Associate",
+         lty=c(1,2), col= c("hotpink","blue"))
```



4 Computing Descriptive Statistics Using Kaplan-Meier Estimates of the Survival Function

My functions `descip()` and `tableStat()` know to treat `"Surv"` objects different from complete (uncensored) measurements. Hence, calls to those functions will compute restricted means, standard deviations, and geometric means using the appropriate area under a survival curve. And those functions will compute quantiles and the probability of exceeding specified thresholds using the Kaplan-Meier estimates.

For instance, if you wanted to obtain the usual descriptive statistics (quartiles are included by default) along with the probability of surviving 8 or 9 years without being promoted for strata defined by sex, you could just execute

```
> descrip(SALARY$ttofull, strata= SALARY$sex, above= c(8,9),
+         subset=SALARY$year==95)

          N     Msng  Restrict  Mean      Std Dev   Geom Mn    Min
All       1597   990 (R 18.00)   9.309+    5.203+    7.957+   0.0000+
```

```
  Str  F   409   225 (R 16.00)   9.445+    4.317+    8.485+   0.0000+
  Str  M  1188   765 (R 18.00)   9.081+    5.262+    7.680+   0.0000+
          25%       Mdn       75%       Max      Pr>8      Pr>9
All       5.000     7.000     13.00     18.00+    0.4060    0.3688
  Str  F  6.000     8.000     13.00     16.00+    0.4215    0.3832
  Str  M  5.000     7.000     13.00     18.00+    0.3980    0.3614
```