Adjusted Survival Curves

Terry M Therneau, Cynthia S Crowson, Elizabeth J Atkinson

Jan 2015

1 Introduction

Suppose we want to investigate to what extent some factor influences survival, as an example we might compare the experience of diabetic patients who are using metformin versus those on injected insulin as their primary treatment modality. There is some evidence that metformin has a positive influence, particularly in cancers, but the ascertainment is confounded by the fact that it is a first line therapy: the patients on metformin will on average be younger and have had a diabetes diagnosis for a shorter amount of time than those using insulin. "Young people live longer" is not a particularly novel observation.

The ideal way to test this is with a controlled clinical trial. This is of course not always possible, and assessments using available data that includes and adjusts for such confounders is also needed. There is extensive literature — and debate — on this topic in the areas of modeling and testing. The subtopic of how to create honest survival curve estimates in the presence of confounders is less well known, and is the focus of this note.

Assume that we have an effect of interest, treatment say, and a set of possible confounding variables. Creation a pair of adjusted survival curves has two parts: definition of a reference population for the confounders, and then the computation of estimated curves for that population. There are important choices in both steps. The first, definition of a target, is often not explicitly stated but can be critical. If an outcome differs with age, myocardial infarction say, and two treatments also had age dependent efficacy, then the comparison will depend greatly on whether we are talking about a population of young, middle aged, or older subjects.

The computational step has two main approaches. The first, sometimes known as *marginal* analysis, first reweights the data such that each subgroup's weighted distribution matches that of our population target. An immediate consequence is that all subgroups will be balanced with respect to the confounding variables. We can then proceed with a simple analysis of survival using the reformulated data, ignoring the confounders. The second approach seeks to understand and model the effect of each confounder, with this we can then correct for them. From a comprehensive overall model we can obtain predicted survival curves for any configuration of variables, and from these get predicted overall curves for the reference population. This is often called the *conditional* approach since we are using conditional survival curves given covariates x.

A third but more minor choice is division of the covariates x into effects of interest vs. confounders. For instance, we might want to see separate curves for two treatments, each adjusted for age and sex. The reference population will describe the age and sex distribution.

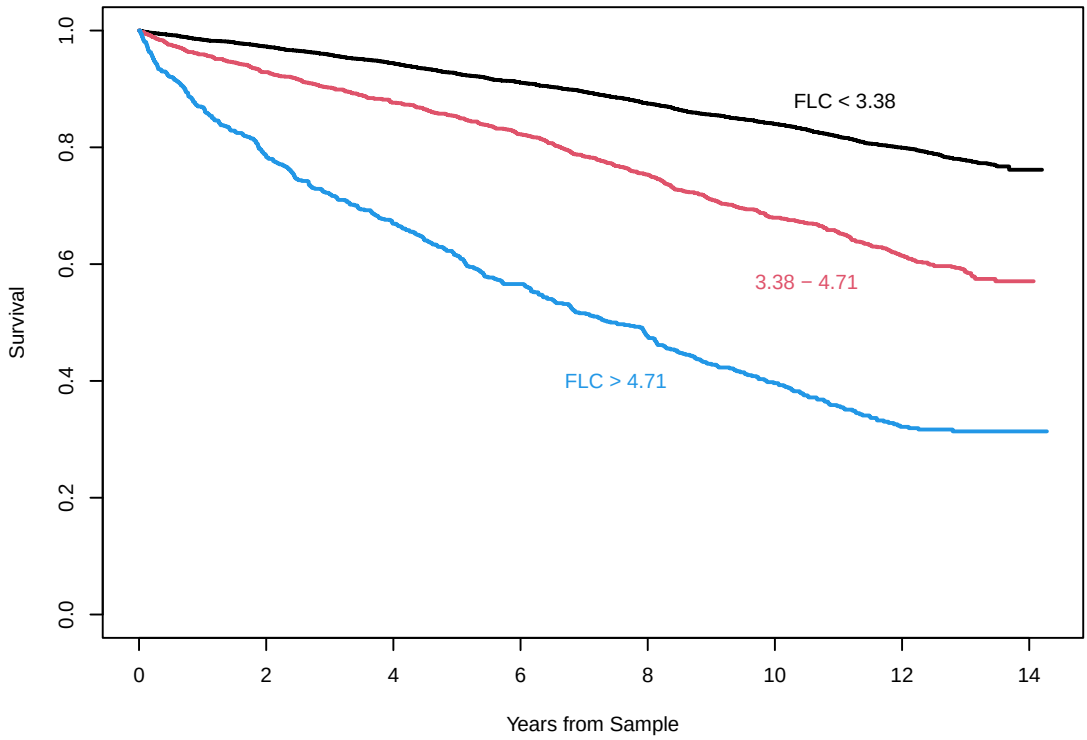


Figure 1: Survival of 7874 residents of Olmsted County, broken into three cohorts based on FLC value.

For simplicity we will use x to describe all the confounding variables and use c for the control variable(s), e.g. treatment. The set c might be empty, producing a single overall curve, but this is the uncommon case. As shown below, our two methods differ essentially in the *order* in which the two necessary operations are done, balancing and survival curve creation.

Marginal:	balance data on x	$\longrightarrow$	form survival curves for each c
Conditional:	predicted curves for $\{x, c\}$ subset	$\longrightarrow$	average the predictions for each c

We can think of them as "balance and then model" versus "model then balance". An analysis might use a combinations of these, of course, balancing on some factors and modeling others. All analyses are marginal analyses with respect to important predictors that are unknown to us, although in that case we have no assurance of balance on those factors.

2 Free Light Chain

Our example data set for this comparison uses a particular assay of plasma immunoglobulins and is based on work of Dr Angela Dispenzieri and her colleagues at the Mayo Clinic [2]. In brief: plasma cells (PC) are responsible for the production of immunoglobulins, but PC comprise only

	50–59	60–69	70–79	80+
FLC < 3.38	2585 (47)	1690 (30)	969 (17)	314 (6)
FLC 3.38–4.71	442 (29)	446 (29)	423 (28)	215 (14)
FLC > 4.71	121 (16)	187 (25)	224 (30)	224 (30)

Table 1: Comparison of the age distributions (percents) for each of the three groups.

a small portion (< 1%) of the total blood and marrow hematapoetic cell population in normal patients. The normal human repertoire is estimated to contain over 10^8 unique immunoglobulins, conferring a broad range of immune protection. In multiple myeloma, the most common form of plasma cell malignancy, almost all of the circulating antigen will be identical, the product of a single malignant clone. An electrophoresis examination of circulating immunoglobulins will exhibit a "spike" corresponding to this unique molecule. This anomaly is used both as a diagnostic method and in monitoring the course of the disease under treatment.

The presence of a similar, albeit much smaller, spike in normal patients has been a long term research interest of the Mayo Clinic hematology research group [5]. In 1995 Dr Robert Kyle undertook a population based study of this, and collected serum samples on 19,261 of the 24,539 residents of Olmsted County, Minnesota, aged 50 years or more [4]. In 2010 Dr. Angela Dispenzieri assayed a sub fraction of the immunoglobulins, the free light chain (FLC), on 15,748 of these subjects who had sufficient remaining sera from the original sample collection. All studies took place under the oversight of the appropriate Institutional Review Boards, which ensure rigorous safety and ethical standards in research.

A subset of the Dispenzieri study is available in the survival package as data set `flchain`. Because the original study assayed nearly the entire population, there is concern that some portions of the anonymized data could be linked to actual subjects by a diligent searcher, and so only a subset of the study has been made available as a measure to strengthen anonymity. It was randomly selected from the whole within sex and age group strata so as to preserve the age/sex structure. The data set contains 3 subjects whose blood sample was obtained on the day of their death. It is rather odd to think of a sample obtained on the final day as "predicting" death, or indeed for any results obtained during a patient's final mortality cascade. There are also a few patients with no follow-up beyond the clinic visit at which the assay occurred. We have chosen in this analysis to exclude the handful of subjects with less than 7 days of follow-up, leaving 7840 observations.

Figure 1 shows the survival curves for three subgroups of the patients: those whose total free light chain (FLC) is in the upper 10% of all values found in the full study, those in the 70–89th percentile, and the remainder. There is a clear survival effect. Average free light chain amounts rise with age, however, at least in part because it is eliminated through the kidneys and renal function declines with age. Table 1 shows the age distribution for each of the three groups. In the highest decile of FLC (group 3) over half the subjects are age 70 or older compared to only 23% in those below the 70th percentile. How much of the survival difference is truly associated with FLC and how much is simply an artifact of age? (The cut points are arbitrary, but we have chosen to mimic the original study and retain them. Division into three groups is a convenient number to illustrate the methods in this vignette, but we do not make any claim that such a categorization is optimal or even sensible statistical practice.) The R code for figure 1 is shown below.

```
> fdata <- flchain[flchain$futime >=7,]
> fdata$age2 <- cut(fdata$age, c(0,54, 59,64, 69,74,79, 89, 110),
                    labels = c(paste(c(50,55,60,65,70,75,80),
                                     c(54,59,64,69,74,79,89), sep='-'), "90+"))
> fdata$group <- factor(1+ 1*(fdata$flc.grp >7) + 1*(fdata$flc.grp >9),
                        levels=1:3,
                        labels=c("FLC < 3.38", "3.38 - 4.71", "FLC > 4.71"))
> sfit1 <- survfit(Surv(futime, death) ~ group, fdata)
> plot(sfit1, mark.time=F, col=c(1,2,4), lty=1, lwd=2,
       xscale=365.25, xlab="Years from Sample",
       ylab="Survival")
> text(c(11.1, 10.5, 7.5)*365.25, c(.88, .57, .4),
       c("FLC < 3.38", "3.38 - 4.71", "FLC > 4.71"), col=c(1,2,4))
```

3 Reference populations

There are a few populations that are commonly used as the reference group.

1. Empirical. The overall distribution of confounders x in the data set as a whole. For this study we would use the observed age/sex distribution, ignoring FLC group. This is also called the "sample" or "data" distribution.
2. External reference. The distribution from some external study or standard.
3. Internal reference. A particular subset of the data is chosen as the reference, and other subsets are then aligned with it.

Method 2 is common in epidemiology, using a reference population based on a large external population such as the age/sex distribution of the 2000 United States census. Method 3 most often arises in the case-control setting, where one group is small and precious (a rare disease say) and the other group (the controls) from which we can sample is much larger. In each case the final result of the computation can be thought of as the expected answer we "would obtain" in a study that was perfectly balanced with respect to the list of confounders x. Population 1 is the most frequent.

4 Marginal approach

4.1 Selection

One approach for balancing is to select a subset of the data such that its distribution matches the referent for each level of c, i.e., for each curve that we wish to obtain. As an example we take a case-control like approach to the FLC data, with FLC high as the "cases" since it is the smallest group. Table 2 shows a detailed distribution of the data with respect to age and sex. The balanced subset has all 32 females aged 50–54 from the high FLC group, a random sample of 32 out of the 738 females in the age 50–54 low FLC group, and 32 out of 110 for the middle FLC. Continue this for all age/sex subsets. We cannot *quite* compute a true case-control estimate for

	Females							
				Age				
FLC group	50–54	55–59	60–64	65–69	70–74	75–79	80–89	90+
Low	738	638	490	442	376	254	224	24
Med	110	94	88	104	116	108	125	21
High	32	30	45	43	48	46	103	32
	Males							
Low	658	553	427	331	227	112	64	2
Med	111	127	124	130	110	90	63	6
High	26	33	39	61	66	65	74	15

Table 2: Detailed age and sex distribution for the study population

this data since there are not enough "controls" in the female 90+ category to be able to select one unique control for each case, and likewise in the male 80-89 and 90+ age groups. To get around this we will sample with replacement in these strata.

```
> temp <- with(fdata, table(group, age2, sex))
> dd <- dim(temp)
> # Select subjects
> set.seed(1978)
> select <- array(vector('list', length=prod(dd)), dim=dd)
> for (j in 1:dd[2]) {
      for (k in 1:dd[3]) {
          n <- temp[3,j,k] # how many to select
          for (i in 1:2) {
              indx <- which(as.numeric(fdata$group)==i &
                            as.numeric(fdata$age2) ==j &
                            as.numeric(fdata$sex) ==k)
              select[i,j,k] <- list(sample(indx, n, replace=(n> temp[i,j,k])))
          }
          indx <- which(as.numeric(fdata$group)==3 &
                        as.numeric(fdata$age2) ==j &
                        as.numeric(fdata$sex) ==k)
          select[3,j,k] <- list(indx)  #keep all the group 3 = high
      }
  }
> data2 <- fdata[unlist(select),]
> sfit2 <- survfit(Surv(futime, death) ~ group, data2)
> plot(sfit2,col=c(1,2,4),  lty=1, lwd=2,
       xscale=365.25, xlab="Years from Sample",
       ylab="Survival")
> lines(sfit1, col=c(1,2,4), lty=2, lwd=1,
        xscale=365.25)
> legend(730, .4, levels(fdata$group), lty=1, col=c(1,2,4),
```

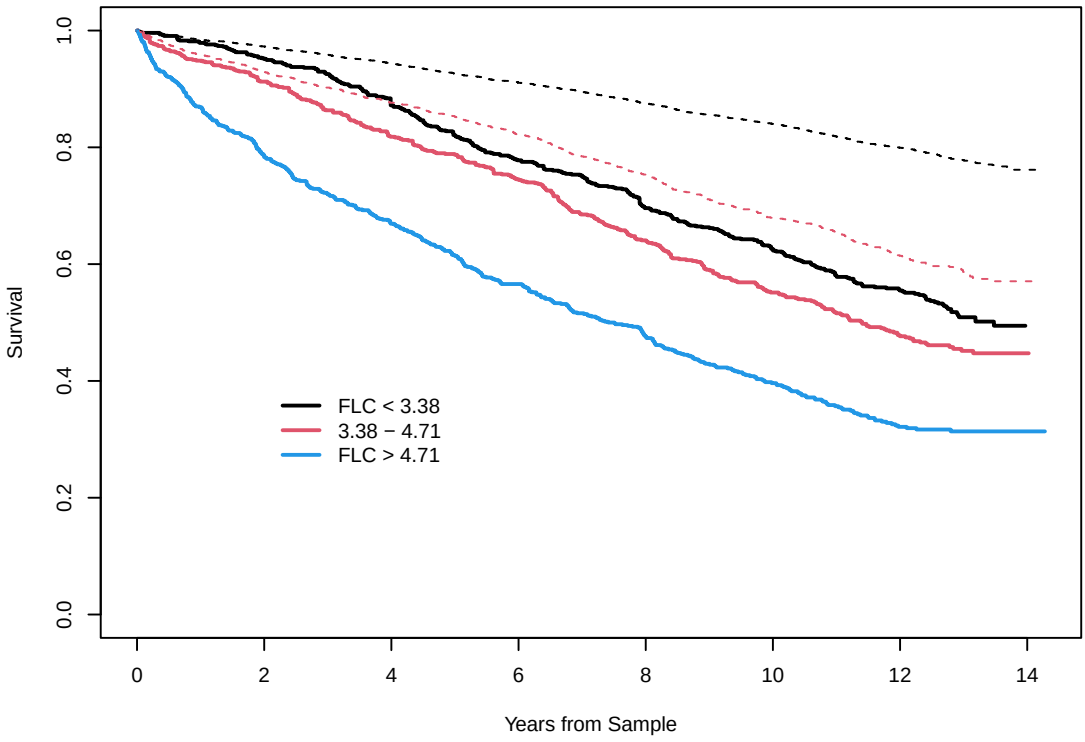


Figure 2: Survival curves from a case-control sample are shown as solid lines, dashed lines are curves for the unweighted data set (as found in figure 1).

```
                    bty='n', lwd=2)
```

The survival curves for the subset data are shown in figure 2. The curve for the high risk group is unchanged, since by definition all of those subjects were retained. We see that adjustment for age and sex has reduced the apparent survival difference between the groups by about half, but a clinically important effect for high FLC values remains. The curve for group 1 has moved more than that for group 2 since the age/sex adjustment is more severe for that group.

In actual practice, case-control designs arise when matching and selection can occur *before* data collection, leading to a substantial decrease in the amount of data that needs to be gathered and a consequent cost or time savings. When a data set is already in hand it has two major disadvantages. The first is that the approach wastes data; throwing away information in order to achieve balance is always a bad idea. Second is that though it returns an unbiased comparison, the result is for a very odd reference population.

One advantage of matched subsets is that standard variance calculations for the curves are correct; the values provided by the usual Kaplan-Meier program need no further processing. We can also use the usual statistical tests to check for differences between the curves.

```
> survdiff(Surv(futime, death) ~ group, data=data2)
```

```
Call:
survdiff(formula = Surv(futime, death) ~ group, data = data2)

                    N Observed Expected (O-E)^2/E (O-E)^2/V
group=FLC < 3.38  758      341      456     29.17     47.37
group=3.38 - 4.71 758      377      415      3.54      5.43
group=FLC > 4.71  758      477      323     73.08    100.71

 Chisq= 106  on 2 degrees of freedom, p= <2e-16
```

4.2 Reweighting

A more natural way to adjust the data distribution is by weighting. Let $\pi(a, s)$, a = age group, s = sex be a target population age/sex distribution for our graph, and $p(a, s, i)$ the observed probability of each age/sex/group combination in the data. Both π and p sum to 1. Then if each observation in the data set is given a case weight of

$$w_{asi} = \frac{\pi(a, s)}{p(a, s, i)} \tag{1}$$

the weighted age/sex distribution for each of the groups will equal the target distribution π. An obvious advantage of this approach is that the resulting curves represent a tangible and well defined group.

As an example, we will first adjust our curves to match the age/sex distribution of the 2000 US population, a common reference target in epidemiology studies. The `uspop2` data set is found in later releases of the survival package in R. It is an array of counts with dimensions of age, sex, and calendar year. We only want ages of 50 and over, and the population data set has collapsed ages of 100 and over into a single category. We create a table `tab100` of observed age/sex counts within group for our own data, using the same upper age threshold. New weights are the values π/p = `pi.us/tab100`.

```
> refpop <- uspop2[as.character(50:100),c("female", "male"), "2000"]
> pi.us  <- refpop/sum(refpop)
> age100 <- factor(ifelse(fdata$age >100, 100, fdata$age), levels=50:100)
> tab100 <- with(fdata, table(age100, sex, group))/ nrow(fdata)
> us.wt  <- rep(pi.us, 3)/ tab100  #new weights by age,sex, group
> range(us.wt)
[1] 0.7751761       Inf
```

There are infinite weights! This is because the US population has coverage at all ages, but our data set does not have representatives in every age/sex/FLC group combination; there are for instance no 95 year old males in in the data set. Let us repeat the process, collapsing the US population from single years into the 8 age groups used previously in table 2. Merging the per age/sex/group weights found in the 3-dimensional array `us.wt` into the data set as per-subject weights uses matrix subscripts, a useful but less known feature of R.

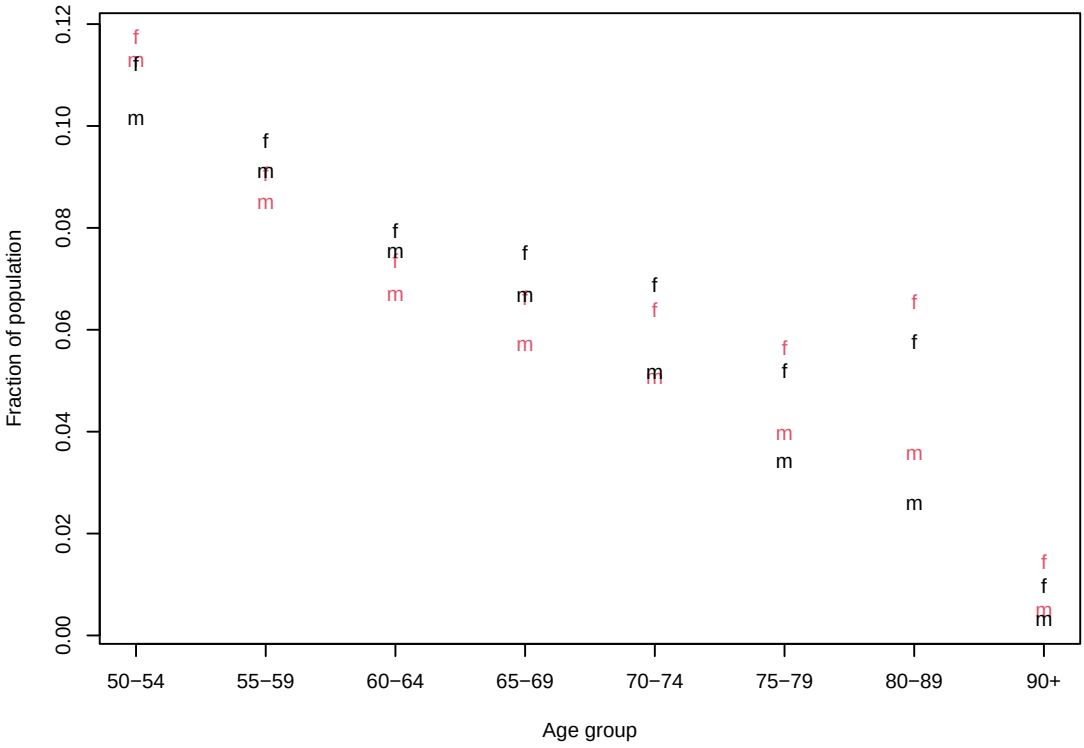


Figure 3: Population totals for the US reference (red) and for the observed data set (black).

```
> temp <- as.numeric(cut(50:100, c(49, 54, 59, 64, 69, 74, 79, 89, 110)+.5))
> pi.us<- tapply(refpop, list(temp[row(refpop)], col(refpop)), sum)/sum(refpop)
> tab2 <- with(fdata, table(age2, sex, group))/ nrow(fdata)
> us.wt <- rep(pi.us, 3)/ tab2
> range(us.wt)
[1]  1.113898 34.002834
> index <- with(fdata, cbind(as.numeric(age2), as.numeric(sex),
                             as.numeric(group)))
> fdata$uswt <- us.wt[index]
> sfit3a <-survfit(Surv(futime, death) ~ group, data=fdata, weights=uswt)
```

A more common choice is to use the overall age/sex distribution of the sample itself as our target distribution π, i.e., the empirical distribution. Since FLC data set is population based and has excellent coverage of the county, this will not differ greatly from the US population in this case, as is displayed in figure 3.

```
> tab1 <- with(fdata, table(age2, sex))/ nrow(fdata)
> matplot(1:8, cbind(pi.us, tab1), pch="fmfm", col=c(2,2,1,1),
          xlab="Age group", ylab="Fraction of population",
```

```
          xaxt='n')
> axis(1, 1:8, levels(fdata$age2))
> tab2 <- with(fdata, table(age2, sex, group))/nrow(fdata)
> tab3 <- with(fdata, table(group)) / nrow(fdata)
> rwt <- rep(tab1,3)/tab2
> fdata$rwt <- rwt[index]  # add per subject weights to the data set
> sfit3 <- survfit(Surv(futime, death) ~ group, data=fdata, weights=rwt)
> temp <- rwt[,1,]   #show female data
> temp <- temp %*% diag(1/apply(temp,2,min))
> round(temp, 1) #show female data
age2    [,1] [,2] [,3]
  50-54  1.0  2.2 11.4
  55-59  1.0  2.2 10.6
  60-64  1.1  2.0  5.8
  65-69  1.1  1.6  5.7
  70-74  1.2  1.3  4.7
  75-79  1.3  1.0  3.7
  80-89  1.7  1.0  1.8
  90+    2.7  1.0  1.0


> plot(sfit3, mark.time=F, col=c(1,2,4), lty=1, lwd=2,
       xscale=365.25, xlab="Years from Sample",
       ylab="Survival")
> lines(sfit3a,  mark.time=F, col=c(1,2,4), lty=1, lwd=1,
        xscale=365.25)
> lines(sfit1,  mark.time=F, col=c(1,2,4), lty=2, lwd=1,
       xscale=365.25)
> legend(730, .4, levels(fdata$group), lty=1, col=c(1,2,4),
                bty='n', lwd=2)
```

The calculation of weights is shown above, and finishes with a table of the weights for the females. The table was scaled so as to have a minimum weight of 1 in each column for simpler reading. We see that for the low FLC group there are larger weights for the older ages, whereas the high FLC group requires substantial weights for the youngest ages in order to achieve balance. The resulting survival curve is shown in figure 4. The distance between the adjusted curves is similar to the results from subset selection, which is as expected since both approaches are correcting for the same bias, but results are now for an overall population distribution that matches Olmsted County. The curves estimate what the results would have looked like, had each of the FLC groups contained the full distribution of ages.

Estimation based on reweighted data is a common theme in survey sampling. Correct standard errors for the curves are readily computed using methods from that literature, and are available in some software packages. In R the `svykm` routine in the `survey` package handles both this simple case and more complex sampling schemes. Tests of the curves can be done using a weighted Cox model; the robust variance produced by `coxph` is identical to the standard Horvitz-Thompsen variance estimate used in survey sampling [1]. The robust score test from

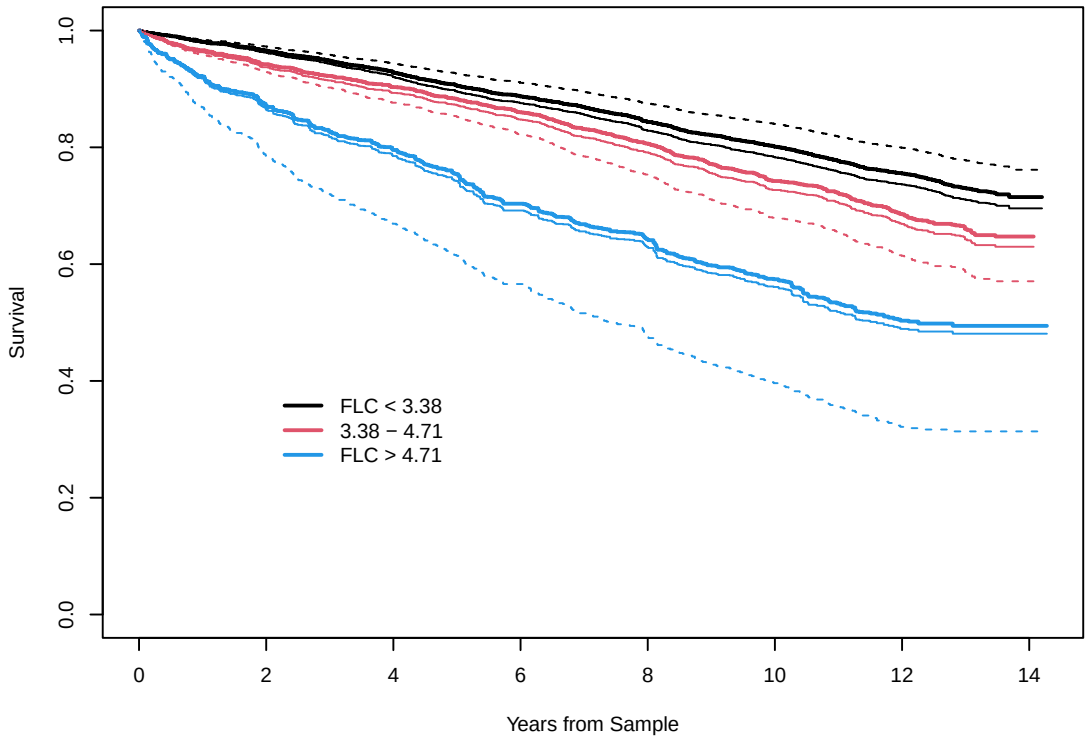


Figure 4: Survival curves for the three groups using reweighted data are shown with solid lines, the original unweighted analysis as dashed lines. The heavier solid line adjusts to the Olmsted population and the lighter one to the US population.

`coxph` corresponds to a log-rank test corrected for weighting. (In the example below the svykm function is only run if the survey package is already loaded, as the variance calculation is very slow for this large data set.)

```
> id <- 1:nrow(fdata)
> cfit <- coxph(Surv(futime, death) ~ group, data=fdata, cluster=id,
                weights=rwt)
> summary(cfit)$robscore
        test           df       pvalue
1.076737e+02 2.000000e+00 4.158712e-24
> if (exists("svykm")) { #true if the survey package is loaded
      sdes <- svydesign(id = ~0, weights=~rwt, data=fdata)
      dfit <- svykm(Surv(futime, death) ~ group, design=sdes, se=TRUE)
  }
```

Note: including the `cluster` term in the coxph call causes it to treat the weights as resampling values and thus use the proper survey sampling style variance. The default without that term

would be to treat the case weights as replication counts. This same alternate variance estimate is also called for when there are correlated observations; many users will be more familiar with the cluster statement in that context.

Inverse probability weighting Notice that when using the overall population as the target distribution π we can use Bayes rule to rewrite the weights as

$$\begin{aligned} \frac{1}{w_{asi}} &= \frac{\Pr(\text{age} = a, \text{sex} = s, \text{group} = i)}{\Pr(\text{age} = a, \text{sex} = s)} \\ &= \Pr(\text{group} = i | \text{age} = a, \text{sex} = s) \end{aligned}$$

This last is precisely the probability estimated by a logistic regression model, leading to *inverse probability weighting* as an alternate label for this approach. We can reproduce the weights calculated just above with three logistic regression models.

```
> options(na.action="na.exclude")
> gg <- as.numeric(fdata$group)
> lfit1 <- glm(I(gg==1) ~ factor(age2) * sex, data=fdata,
              family="binomial")
> lfit2 <- glm(I(gg==2) ~ factor(age2) * sex, data=fdata,
              family="binomial")
> lfit3 <- glm(I(gg==3) ~ factor(age2) * sex, data=fdata,
              family="binomial")
> temp <- ifelse(gg==1, predict(lfit1, type='response'),
                 ifelse(gg==2, predict(lfit2, type='response'),
                        predict(lfit3, type='response')))
> all.equal(1/temp, fdata$rwt)
[1] TRUE
```

If there were only 2 groups then only a single regression model is needed since P(group 2) = 1 - P(group 1). Note the setting of na.action, which causes the predicted vector to have the same length as the original data even when there are missing values. This simplifies merging the derived weights with the original data set.

An advantage of the regression framework is that one can easily accommodate more variables by using a model with additive terms and only a few selected interactions, and the model can contain continuous as well as categorical predictors. The disadvantage is that such models are often used without the necessary work to check their validity. For instance models with `age + sex` could have been used above. This makes the assumption that the odds of being a member of group 1 is linear in age and with the same slope for males and females; ditto for the models for group 2 and group 3. How well does this work? Since the goal of reweighting is to standardize the ages, a reasonable check is to compute and plot the reweighted age distribution for each flc group.

Figure 5 shows the result. The reweighted age distribution is not perfectly balanced, i.e., the '1', '2' and '3' symbols do no exactly overlay one another, but in this case the simple linear model has done an excellent job. We emphasize that whenever the reweighting is based on a simplified model then such a check is obligatory. It is quite common that a simple model is

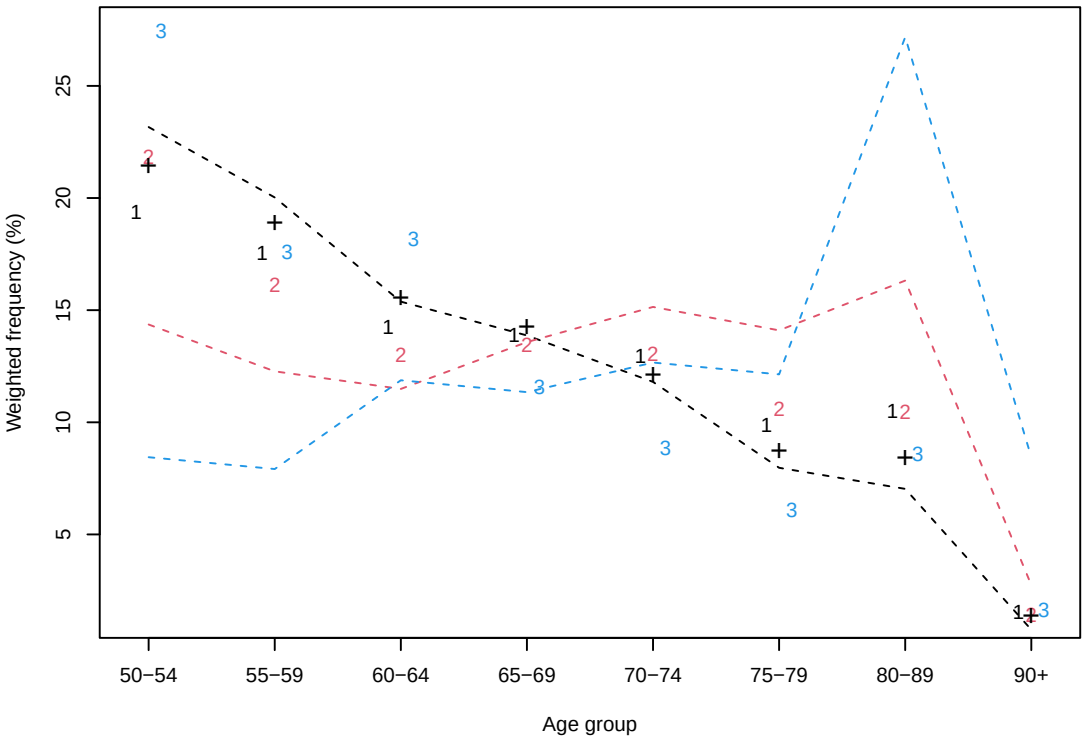


Figure 5: The re-weighted age distribution using logistic regression with continuous age, for females, FLC groups 1–3. The target distribution is shown as a "+". The original unadjusted distribution is shown as dashed lines.

not sufficient and the resulting weight adjustment is inadequate. Like a duct tape auto repair, proceeding forward as though the underlying problem has been addressed is then most unwise.

```
> lfit1b <-glm(I(gg==1) ~ age + sex, data=fdata,
              family="binomial")
> lfit2b <- glm(I(gg==2) ~ age +sex, data=fdata,
              family="binomial")
> lfit3b <- glm(I(gg==3) ~ age + sex, data=fdata,
              family="binomial")
> # weights for each group using simple logistic
> twt <- ifelse(gg==1, 1/predict(lfit1b, type="response"),
               ifelse(gg==2, 1/predict(lfit2b, type="response"),
                                      1/predict(lfit3b, type="response")))
> tdata <- data.frame(fdata, lwt=twt)
> #grouped plot for the females
> temp <- tdata[tdata$sex=='F',]
> temp$gg <- as.numeric(temp$group)
```

```
> c1 <- with(temp[temp$gg==1,], tapply(lwt, age2, sum))
> c2 <- with(temp[temp$gg==2,], tapply(lwt, age2, sum))
> c3 <- with(temp[temp$gg==3,], tapply(lwt, age2, sum))
> xtemp <- outer(1:8, c(-.1, 0, .1), "+") #avoid overplotting
> ytemp <- 100* cbind(c1/sum(c1), c2/sum(c2), c3/sum(c3))
> matplot(xtemp, ytemp, col=c(1,2,4),
          xlab="Age group", ylab="Weighted frequency (%)", xaxt='n')
> ztab <- table(fdata$age2)
> points(1:8, 100*ztab/sum(ztab), pch='+', cex=1.5, lty=2)
> # Add the unadjusted
> temp <- tab2[,1,]
> temp <- scale(temp, center=F, scale=colSums(temp))
> matlines(1:8, 100*temp, pch='o', col=c(1,2,4), lty=2)
> axis(1, 1:8, levels(fdata$age2))
```

Rescaled weights As the weights were defined in equation 1, the sum of weights for each of the groups is 7845, the number of observations in the data set. Since the number of subjects in group 3 is one seventh of that in group 1, the average weight in group 3 is much larger. An alternative is to define weights in terms of the *within* group distribution rather than the overall distribution, leading to the rescaled weights w^*

$$w^* = \frac{\pi(a,s)}{p(a,s|i)} \tag{2}$$

$$= \frac{\mathrm{P}(\text{group} = i)}{\mathrm{P}(\text{group} = i|\text{age} = a, \text{sex} = s)} \tag{3}$$

Each group's weights are rescaled by the overall prevalence of the group. In its simplest form, the weights in each group are scaled to add up to the number of subjects in the group.

```
> # compute new weights
> wtscale <- table(fdata$group)/ tapply(fdata$rwt, fdata$group, sum)
> wt2 <- c(fdata$rwt * wtscale[fdata$group])
> c("rescaled cv"= sd(wt2)/mean(wt2), "rwt cv"=sd(fdata$rwt)/mean(fdata$rwt))
rescaled cv      rwt cv
  0.3612141   1.2772960
> cfit2a <- coxph(Surv(futime, death) ~ group, cluster = id,
                  data=fdata, weights= rwt)
> cfit2b <- coxph(Surv(futime, death) ~ group, cluster = id,
                  data=fdata, weights=wt2)
> round(c(cfit2a$rscore, cfit2b$rscore),1)
[1] 107.7 116.3
```

The rescaling results in weights that are much less variable across groups. This operation has no impact on the individual survival curves or their standard errors, since within group we have multiplied all weights by a constant. When comparing curves across groups, however, the rescaled weights reduce the standard error of the test statistic. This results in increased power for the robust score test, although in this particular data set the improvement is not very large.

5 Conditional method

In the marginal approach we first balance the data set and then compute results on the adjusted data. In the conditional approach we first compute a predicted survival curve for each subject that accounts for flc group, age and sex, and then take a weighted average of the curves to get an overall estimate for each flc group. For both methods a central consideration is the population of interest, which drives the weights. Modeling has not removed the question of *who* these curves should represent, it has simply changed the order of operation between the weighting step and the survival curves step.

5.1 Stratification

Our first approach is to subset the data into homogeneous age/sex strata, compute survival curves within each strata, and then combine results. We will use the same age/sex combinations as before. The interpretation of these groups is different, however. In the marginal approach it was important to find age/sex groups for which the probability of membership within each FLC group was constant within the strata (independent of age and sex, within strata), in this case it is important that the survival for each FLC group is constant in each age/sex stratum. Homogeneity of membership within each stratum and homogeneity of survival within each stratum may lead to different partitions for some data sets.

Computing curves for all the combinations is easy.

```
> allfit <- survfit(Surv(futime, death) ~ group +
                                  age2 + sex, fdata)
> temp <- summary(allfit)$table
> temp[1:6, c(1,4)] #abbrev printout to fit page
                                       records events
group=FLC < 3.38, age2=50-54, sex=F        738     28
group=FLC < 3.38, age2=50-54, sex=M        658     45
group=FLC < 3.38, age2=55-59, sex=F        638     52
group=FLC < 3.38, age2=55-59, sex=M        553     58
group=FLC < 3.38, age2=60-64, sex=F        490     52
group=FLC < 3.38, age2=60-64, sex=M        427     63
```

The resultant survival object has 48 curves: 8 age groups * 2 sexes * 3 FLC groups. To get a single curve for the first FLC group we need to take a weighted average over the 16 age/sex combinations that apply to that group, and similarly for the second and third FLC subset. Combining the curves is a bit of a nuisance computationally because each of them is reported on a different set of time points. A solution is to use the `summary` function for survfit objects along with the `times` argument of that function. This feature was originally designed to allow printout of curves at selected time points (6 months, 1 year, ...), but can also be used to select a common set of time points for averaging. We will arbitrarily use 4 per year, which is sufficient to create a visually smooth plot over the time span of interest. By default `summary` does not return data for times beyond the end of a curve, i.e., when there are no subjects left at risk; the `extend` argument causes a full set of times to always be reported. As seen in the printout above, the computed curves are in sex within age within group order. The overall curve is a weighted average chosen to match the original age/sex distribution of the population.

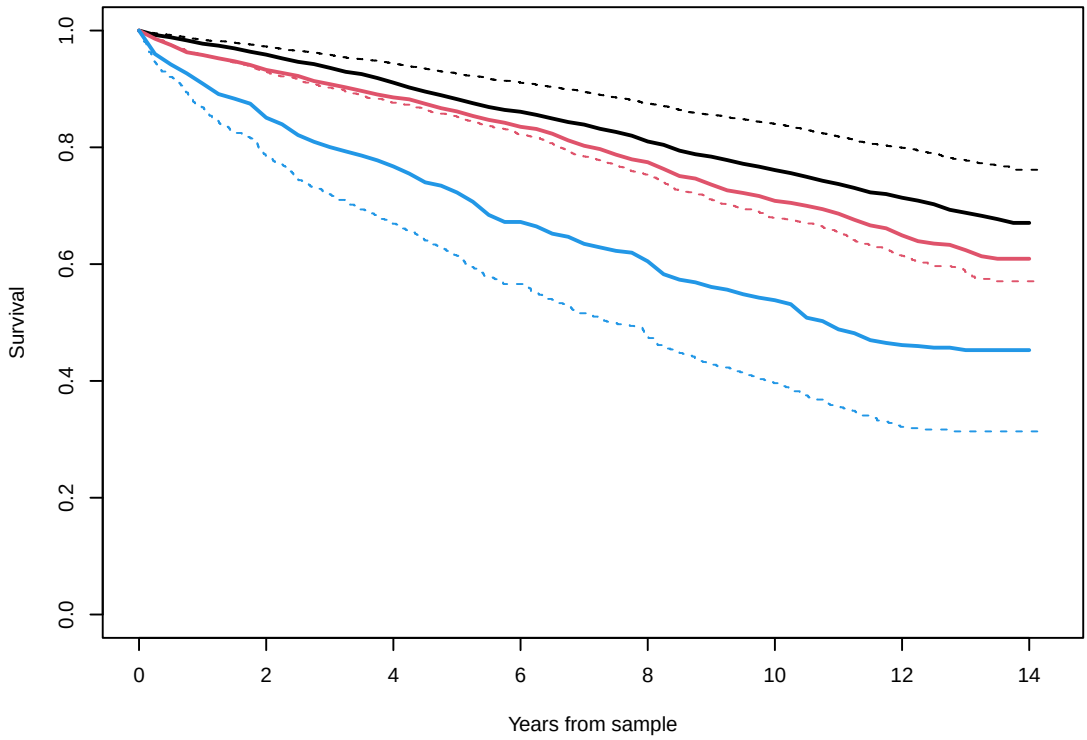


Figure 6: Estimated curves from a stratified model, along with those from the uncorrected fit as dashed lines.

```
> xtime <- seq(0, 14, length=57)*365.25  #four points/year for 14 years
> smat <- matrix(0, nrow=57, ncol=3) # survival curves
> serr <- smat  #matrix of standard errors
> pi <- with(fdata, table(age2, sex))/nrow(fdata)  #overall dist
> for (i in 1:3) {
      temp <- allfit[1:16 + (i-1)*16] #curves for group i
      for (j in 1:16) {
          stemp <- summary(temp[j], times=xtime, extend=T)
          smat[,i] <- smat[,i] + pi[j]*stemp$surv
          serr[,i] <- serr[,i] + pi[j]*stemp$std.err^2
          }
      }
> serr <- sqrt(serr)
> plot(sfit1, lty=2, col=c(1,2,4), xscale=365.25,
       xlab="Years from sample", ylab="Survival")
> matlines(xtime, smat, type='l', lwd=2, col=c(1,2,4),lty=1)
```

Figure 6 shows the resulting averaged curves. Overlaid are the curves for the unadjusted model.

Very careful comparison of these curves with the weighted estimate shows that they have almost identical spread, with just a tiny amount of downward shift.

There are two major disadvantages to the stratified curves. The first is that when the original data set is small or the number of confounders is large, it is not always feasible to stratify into a large enough set of groups that each will be homogeneous. The second is a technical aspect of the standard error estimate. Since the curves are formed from disjoint sets of observations they are independent and the variance of the weighted average is then a weighted sum of variances. However, when a Kaplan-Meier curve drops to zero the usual standard error estimate at that point involves 0/0 and becomes undefined, leading to the NaN (not a number) value in R. Thus the overall standard error becomes undefined if any of the component curves falls to zero. In the above example this happens at about the half way point of the graph. (Other software packages carry forward the se value from the last no-zero point on the curve, but the statistical validity of this is uncertain.)

To test for overall difference between the curves we can use a stratified test statistic, which is a sum of the test statistics computed within each subgroup. The most common choice is the stratified log-rank statistic which is shown below. The score test from a stratified Cox model would give the same result.

```
> survdiff(Surv(futime, death) ~ group + strata(age2, sex), fdata)
Call:
survdiff(formula = Surv(futime, death) ~ group + strata(age2,
    sex), data = fdata)

                     N Observed Expected (O-E)^2/E
group=FLC < 3.38  5560     1110     1360     46.07
group=3.38 - 4.71 1527      562      531      1.77
group=FLC > 4.71   758      477      257    187.59
                  (O-E)^2/V
group=FLC < 3.38     138.22
group=3.38 - 4.71      2.42
group=FLC > 4.71     230.34

 Chisq= 261  on 2 degrees of freedom, p= <2e-16
```

5.2 Modeling

The other approach for conditional estimation is to model the risks due to the confounders. Though we have left it till last, this is usually the first (and most often the only) approach used by most data analysts.

Let's start with the very simplest method: a stratified Cox model.

```
> cfit4a <- coxph(Surv(futime, death) ~ age + sex + strata(group),
                  data=fdata)
> surv4a <- survfit(cfit4a)
> plot(surv4a, col=c(1,2,4), mark.time=F, xscale=365.25,
       xlab="Years post sample", ylab="Survival")
```

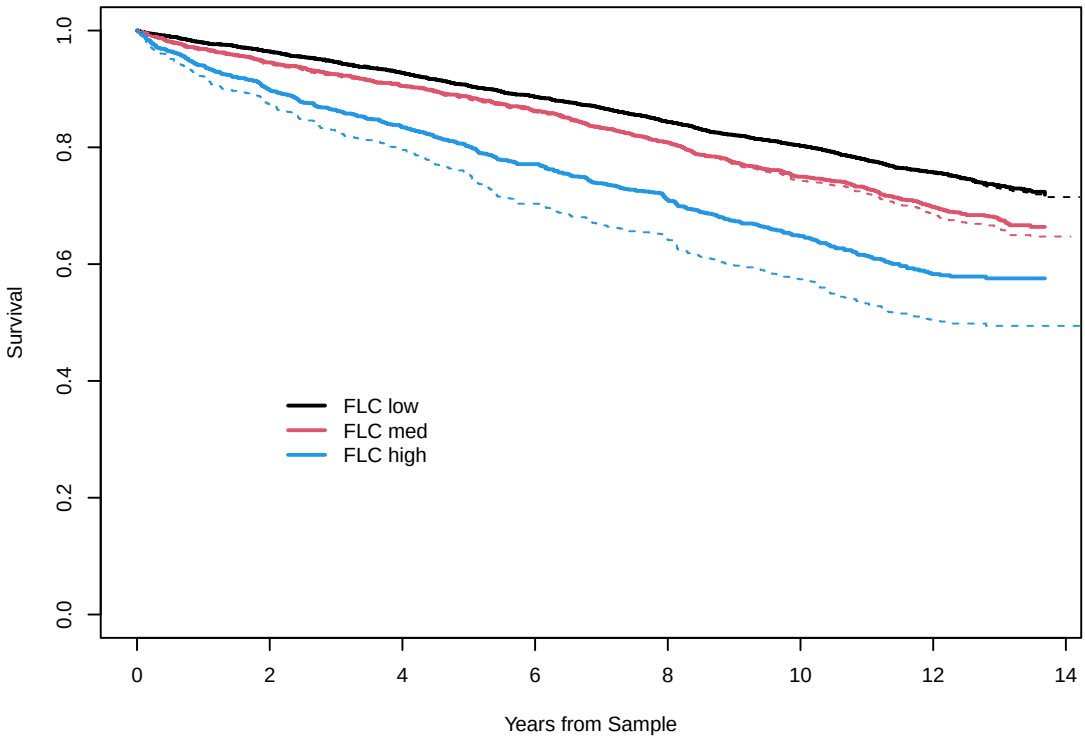


Figure 7: Curves for the three groups, adjusted for age and sex via a risk model. Dotted lines show the curves from marginal adjustment. Solid curves are for the simple risk model `cfit4a`.

This is a very fast and easy way to produce a set of curves, but it has three problems. First is the assumption that this simple model adequately accounts for the effects of age and sex on survival. That is, it assumes that the effect of age on mortality is linear, the sex difference is constant across all ages, and that the coefficients for both are identical for the three FLC groups. The second problem with this approach is that it produces the predicted curve for a single hypothetical subject of age 64.3 years and sex 0, the means of the covariates, under each of the 3 FLC scenarios. However, we are interested in the adjusted survival of a *cohort* of subjects in each range of FLC, and the survival of an "average" subject is not the average survival of a cohort. The third and most serious issue is that it is not clear exactly what these "adjusted" curves represent — exactly who *is* this subject with a sex of 0? Multiple authors have commented on this problem, see Thomsen et al [8], Nieto and Coresh [6] or chapter 10 of Therneau and Grambsh [7] for examples. Even worse is a Cox model that treated the FLC group as a covariate, since that will impose an additional constraint of proportional hazards across the 3 FLC groups.

We can address this last problem problem by doing a proper average. A Cox model fit can produce the predicted curves for any age/sex combination. The key idea is to produce a predicted survival curve for every subject of some hypothetical population, and then take the average of these curves. The most straightforward approach is to retrieve the predicted individual curves

for all 7845 subjects in the data set, assuming each of the three FLC strata one by one, and take a simple average for each strata. For this particular data set that is a bit slow since it involves 7845 curves. However there are only 98 unique age/sex pairs in the data, it is sufficient to obtain the 98 * 3 FLC groups unique curves and take a weighted average. We will make use of the survexp function, which is designed for just this purpose. Start by creating a data set which has one row for each age/sex combination along with its count. Then replicate it into 3 copies, assigning one copy to each of the three FLC strata.

```
> tab4a <- with(fdata, table(age, sex))
> uage <- as.numeric(dimnames(tab4a)[[1]])
> tdata <- data.frame(age = uage[row(tab4a)],
                      sex = c("F","M")[col(tab4a)],
                      count= c(tab4a))
> tdata3 <- tdata[rep(1:nrow(tdata), 3),]  #three copies
> tdata3$group <- factor(rep(1:3, each=nrow(tdata)),
                         labels=levels(fdata$group))
> sfit4a <- survexp(~group, data=tdata3, weight = count,
                      ratetable=cfit4a)
> plot(sfit4a, mark.time=F, col=c(1,2,4), lty=1, lwd=2,
       xscale=365.25, xlab="Years from Sample",
       ylab="Survival")
> lines(sfit3,  mark.time=F, col=c(1,2,4), lty=2, lwd=1,
        xscale=365.25)
> legend(730,.4, c("FLC low", "FLC med", "FLC high"), lty=1, col=c(1,2,4),
                bty='n', lwd=2)
```

Figure 7 shows the result. Comparing this to the prior 3 adjustments shown in figures 4, 5, and 6 we see that this result is different. Why? Part of the reason is due to the fact that $E[f(X)] \neq f(E[X])$ for any non-linear operation f, so that averages of survival curves and survival curves of averages will never be exactly the same. This may explain the small difference between the stratified and the marginal approaches of figures 4 and 6, which were based on the same subsets. The Cox based result is systematically higher than the stratified one, however, so something more is indicated.

Aside: An alternate computational approach is to create the individual survival curves using the `survfit` function and then take averages.

```
> tfit <- survfit(cfit4a, newdata=tdata, se.fit=FALSE)
> curves <- vector('list', 3)
> twt <- c(tab4a)/sum(tab4a)
> for (i in 1:3) {
      temp <- tfit[i,]
      curves[[i]] <- list(time=temp$time, surv= c(temp$surv %*% twt))
      }
```

The above code is a bit sneaky. I know that the result from the survfit function contains a matrix `tfit$surv` of 104 columns, one for each row in the tdata data frame, each column containing the curves for the three strata one after the other. Sub setting `tfit` results in the matrix for a single flc group. Outside of R an approach like the above may be needed, however.

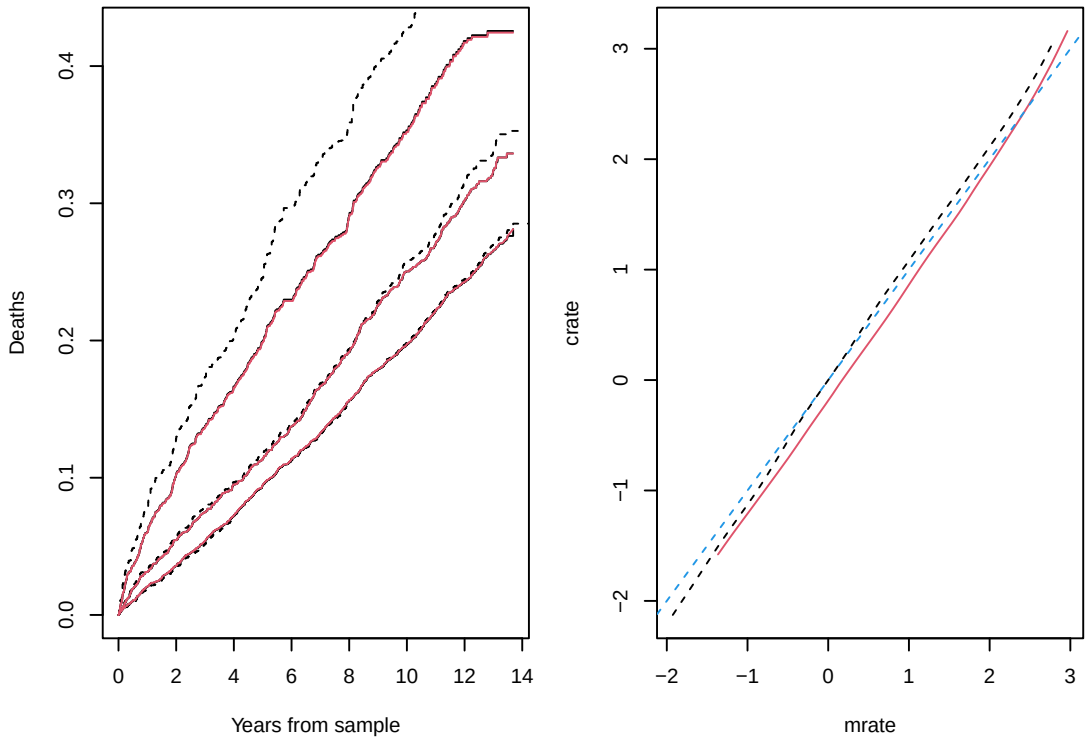


Figure 8: Left panel: comparison of Cox model based adjustment (solid) with the curves based on marginal adjustment (dashed). The Cox model curves without (black) and with (red) an age*sex interaction term overlay. Right panel: plot of the predicted relative risks from a Cox model `crate` versus population values from the Minnesota rate table.

So why are the modeling results so different than either reweighting or stratification? Suspicion first falls on the use of a simple linear model for age and sex, so start by fitting a slightly more refined model that allows for a different slope for the two sexes, but is still linear in age. In this particular data set an external check on the fit is also available via the Minnesota death rate tables, which are included with the survival package as `survexp.mn`. This is an array that contains daily death rates by age, sex, and calendar year.

```
> par(mfrow=c(1,2))
> cfit4b <- coxph(Surv(futime, death) ~ age*sex + strata(group),
                  fdata)
> sfit4b <- survexp(~group, data=tdata3, ratetable=cfit4b, weights=count)
> plot(sfit4b, fun='event', xscale=365.25,
       xlab="Years from sample", ylab="Deaths")
> lines(sfit3, mark.time=FALSE, fun='event', xscale=365.25, lty=2)
> lines(sfit4a, fun='event', xscale=365.25, col=2)
> temp <- median(fdata$sample.yr)
> mrate <- survexp.mn[as.character(uage),, as.character(temp)]
> crate <- predict(cfit4b, newdata=tdata, reference='sample', type='lp')
> crate <- matrix(crate, ncol=2)[,2:1] # mrate has males then females, match it
> # crate contains estimated log(hazards) relative to a baseline,
> #  and mrate absolute hazards, make both relative to a 70 year old
> for (i in 1:2) {
      mrate[,i] <- log(mrate[,i]/ mrate[21,2])
      crate[,i] <- crate[,i] - crate[21,2]
      }
> matplot(mrate, crate, col=2:1, type='l')
> abline(0, 1, lty=2, col=4)
```

The resulting curves are shown in the left panel of figure 8 and reveal that addition of an interaction term did not change the predictions, and that the Cox model result for the highest risk group is distinctly different predicted survival for the highest FLC group is distinctly different when using model based prediction. The right hand panel of the figure shows that though there are slight differences with the Minnesota values, linearity of the age effect is very well supported. So where exactly does the model go wrong? Since this is such a large data set we have the luxury of looking at subsets. This would be a very large number of curves to plot — age by sex by FLC = 48 — so an overlay of the observed and expected curves by group would be too confusing. Instead we will summarize each of the groups according to their observed and predicted number of events.

```
> obs <- with(fdata, tapply(death, list(age2, sex, group), sum))
> pred<- with(fdata, tapply(predict(cfit4b, type='expected'),
                list(age2, sex, group), sum))
> excess <- matrix(obs/pred, nrow=8)  #collapse 3 way array to 2
> dimnames(excess) <- list(dimnames(obs)[[1]], c("low F", "low M",
                                                 "med F", "med M",
                                                 "high F", "high M"))
> round(excess, 1)
```

```
      low F low M med F med M high F high M
50-54   0.9   1.0   1.0   1.0    2.3    2.5
55-59   1.1   0.9   1.6   0.9    1.8    1.6
60-64   0.9   0.8   1.0   1.0    1.5    1.0
65-69   0.8   0.9   0.9   1.1    1.5    1.3
70-74   0.8   1.0   0.8   0.9    1.1    1.2
75-79   1.2   1.0   1.1   1.2    0.8    1.1
80-89   1.2   1.1   1.0   0.9    0.7    0.9
90+     1.3   0.8   1.0   1.6    1.1    0.9
```

The excess risks, defined as the observed/expected number of deaths, are mostly modest ranging from .8 to 1.2. The primary exception exception is the high FLC group for ages 50–59 which has values of 1.6 to 2.5; the Cox model fit has greatly overestimated the survival for the age 50–54 and 55–59 groups. Since this is also the age category with the highest count in the data set, this overestimation will have a large impact on the overall curve for high FLC subset, which is exactly where the the deviation in figure 8 is observed to lie. There is also mild evidence for a linear trend in age for the low FLC females, in the other direction. Altogether this suggests that the model might need to have a different age coefficient for each of the three FLC groups.

```
> cfit5a <- coxph(Surv(futime, death) ~ strata(group):age +sex, fdata)
> cfit5b <- coxph(Surv(futime, death) ~ strata(group):(age +sex), fdata)
> cfit5c <- coxph(Surv(futime, death) ~ strata(group):(age *sex), fdata)
> options(show.signif.stars=FALSE) # see footnote
> anova(cfit4a, cfit5a, cfit5b, cfit5c)
Analysis of Deviance Table
 Cox model: response is  Surv(futime, death)
 Model 1: ~ age + sex + strata(group)
 Model 2: ~ strata(group):age + sex
 Model 3: ~ strata(group):(age + sex)
 Model 4: ~ strata(group):(age * sex)
  loglik   Chisq Df Pr(>|Chi|)
1 -15155
2 -15139 31.0215  2  1.836e-07
3 -15139  0.0615  2    0.96972
4 -15136  6.9116  3    0.07477
> temp <- coef(cfit5a)
> names(temp) <- c("sex", "ageL", "ageM", "ageH")
> round(temp,3)
  sex  ageL  ageM  ageH
0.330 0.112 0.101 0.078
```

The model with separate age coefficients for each FLC group gives a major improvement in goodness of fit, but adding separate sex coefficients per group or further interactions does not add importantly beyond that. [1]

[1] There are certain TV shows that make one dumber just by watching them; adding stars to the output has the same effect on statisticians.

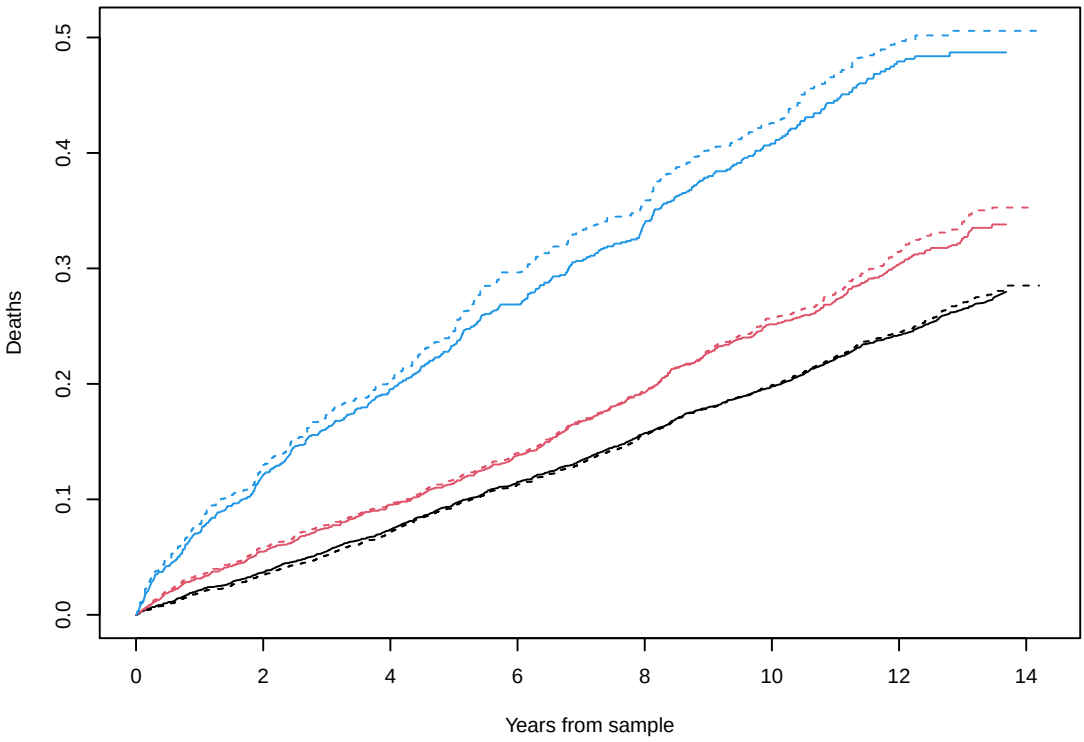


Figure 9: Adjusted survival for the 3 FLC groups based on the improved Cox model fit. Dashed lines show the predictions from the marginal model.

A recheck of the observed/expected values now shows a much more random pattern, though some excess remains in the upper right corner. The updated survival curves are shown in figure 9 and now are in closer concordance with the marginal fit.

```
> pred5a <- with(fdata, tapply(predict(cfit5a, type='expected'),
                list(age2, sex, group), sum))
> excess5a <- matrix(obs/pred5a, nrow=8,
                     dimnames=dimnames(excess))
> round(excess5a, 1)
      low F low M med F med M high F high M
50-54   1.0   1.4   0.9   1.1    1.2    1.5
55-59   1.2   1.1   1.4   0.9    1.1    1.1
60-64   0.9   0.9   1.0   1.1    1.0    0.8
65-69   0.8   1.0   0.9   1.1    1.2    1.1
70-74   0.8   1.0   0.8   0.9    1.0    1.1
75-79   1.2   0.9   1.1   1.1    0.8    1.0
80-89   1.1   0.9   1.0   0.8    0.9    0.9
90+     1.2   0.6   1.1   1.5    1.5    1.1
> sfit5  <- survexp(~group,  data=tdata3, ratetable=cfit5a, weights=count)
> plot(sfit3, fun='event', xscale=365.25, mark.time=FALSE, lty=2, col=c(1,2,4),
       xlab="Years from sample", ylab="Deaths")
> lines(sfit5, fun='event', xscale=365.25, col=c(1,2,4))
```

One problem with the model based estimate is that standard errors for the curves are complex. Standard errors of the individual curves for each age/sex/FLC combination are a standard output of the survfit function, but the collection of curves is correlated since they all depend on a common estimate of the model's coefficient vector β. Curves with disparate ages are anti-correlated (an increase in the age coefficient of the model would raise one and lower the other) whereas those for close ages are positively correlated. A proper variance for the unweighted average has been derived by Gail and Byar [3], but this has not been implemented in any of the standard packages, nor extended to the weighted case. A bootstrap estimate would appear to be the most feasible.

6 Conclusions

When two populations need to be adjusted and one is much larger than the other, the balanced subset method has been popular. It is most often seen in the context of a case-control study, with cases as the rarer group and a set of matched controls selected from the larger one. This method has the advantage that the usual standard error estimates from a standard package are appropriate, so no further work is required. However, in the general situation it leads to a correct answer but for the wrong problem, i.e., not for a population in which we are interested.

The population reweighted estimate is flexible, has a readily available variance in some statistical packages (but not all), and the result is directly interpretable. It is the method we recommend in general. The approach can be extended to a large number of balancing factors by using a regression model to derive the weights. Exploration and checking of said model for

adequacy is an important step in this case. The biggest downside to the method arises when there is a subset which is rare in the data sample but frequent in the adjusting population. In this case subjects in that subset will be assigned large weights, and the resulting curves will have high variance.

The stratified method is closely related to reweighting (not shown). It does not do well if the sample size is small, however.

Risk set modeling is a very flexible method, but is also the one where it is easiest to go wrong by using an inadequate model, and variance estimation is also difficult. To the extent that the fitted model is relevant, it allows for interpolation and extrapolation to a reference population with a different distribution of covariates than the one in the training data. It may be applicable in cases such as rare subsets where population reweighting is problematic, with the understanding that one is depending heavily on extrapolation in this case, which is always dangerous.

7 A note on type 3 tests

One particular software package (not R) and its proponents are very fond of something called "type 3" tests. Said tests are closely tied to a particular reference population:

- For all continuous covariates in the model, the empirical distribution is used as the reference.
- For all categorical adjusters, a uniform distribution over the categories is used.

Figure 10 shows the fit from such a model. Not surprisingly, the predicted death rate is very high: 1/4 of our population is over 80 years old! The authors do not find such a prediction particularly useful since we don't ever expect to see a population like this (it's sort of like planning for the zombie apocalypse), but for those enamored of type 3 tests this shows how to create the corresponding curves.

```
> # there is a spurious warning from the model below: R creates 3 unneeded
> # columns in the X matrix
> cfit6 <- coxph(Surv(futime, death) ~ strata(group):age2 + sex, fdata)
> saspop <- with(fdata, expand.grid(age2= levels(age2), sex= levels(sex),
                                    group = levels(group)))
> sfit6  <- survexp(~group,  data=saspop, ratetable=cfit6)
> plot(sfit6, fun='event', xscale=365.25, mark.time=FALSE, lty=1, col=c(1,2,4),
       xlab="Years from sample", ylab="Deaths")
> lines(sfit5, fun='event', xscale=365.25, lty=2, col=c(1,2,4))
```

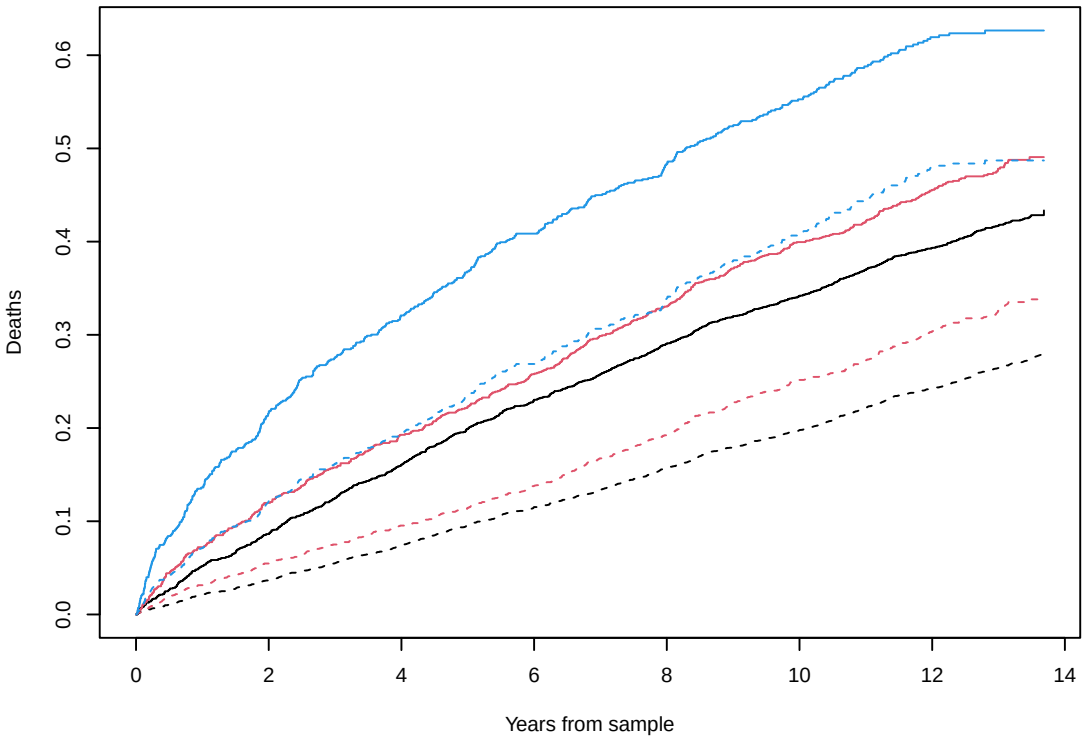


Figure 10: Adjusted survival for the 3 FLC groups based on a fit with categorical age, and predicting for a uniform age/sex population. Dashed lines show the predictions from the marginal model.

References

[1] D. A. Binder. Fitting Cox's proportional hazards models from survey data. *Biometrika*, 79:139–147, 1992.

[2] A. Dispenzieri, J. Katzmann, R. Kyle, D. Larson, T. Therneau, C. Colby, R. Clark, G. Mead, S. Kumar, L.J. Melton III, and S.V. Rajkumar. Use of monclonal serum immunoglobulin free light chains to predict overall survival in the general population. *Mayo Clinic Proc*, 87:512–523, 2012.

[3] M. H. Gail and D. P. Byar. Variance calculations for direct adjusted survival curves, with applications to testing for no treatment effect. *Biometrical J.*, 28:587–599, 1986.

[4] R. Kyle, T. Therneau, S.V. Rajkumar, D. Larson, M. Pleva, J. Offord, A. Dispenzieri, J. Katzman, and L.J. Melton III. Prevalence of monoclonal gammopathy of undetermined significance. *New England J Medicine*, pages 1362–1369, 2006.

[5] R. A. Kyle. "Benign" monoclonal gammopathy — after 20 to 35 years of follow-up. *Mayo Clinic Proceedings*, 68:26–36, 1993.

[6] F. Javier Nieto and Josef Coresh. Adjusting survival curves for confounders: a review and a new method. *Am J of Epidemiology*, pages 1059–1068, 1996.

[7] T. M. Therneau and P. M. Grambsch. *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag, New York, 2000.

[8] B. L. Thomsen, N. Keiding, and D. G. Altman. A note on the calculation of expected survival, illustrated by the survival of liver transplant patients. *Stat. in Medicine*, 10:733–738, 1991.