It is easier, and more useful, to use S rather than $\mathscr{F}$ for the survival function.

The estimator is the product (over i, the index of the distinct failure times occurring before t) of estimated conditional survival probabilities. It is given in Equation 38.

Think of it as $\prod\{1-\hat{\lambda}_{0_i}\}$, where $\hat{\lambda}_{0_i}$ is treated as a *probability* of failure, since we are now viewing getting to t without failing as getting over a (finite) series of hurdles, one hurdle at each distinct observed failure time t_i. Thus, $\hat{\lambda}_{0_i}$ is the estimated conditional probability of failure at hurdle i for persons with covariate vector $z=0$ attempting hurdle i. [We ignore the fact that – artificially, because of the conditioning – this probability cannot be estimated as 0, since we observed at least one failure at the hurdle.]

Unfortunately the data for estimation of $\hat{\lambda}_{0_i}$ consist of a sample of heterogeneous-risk individuals, i.e., they have different values of z. Some may have $z=0$ but most or all (if z is, say, age and cholesterol) will have values a long way from zero (unless the single z is temperature, and we are dealing with measurements taken near the 0 value for the temperature scale!). So, we may not have any observations with $z=0$, and so we must synthetically convert the observations to "$z=0$ - equivalent" observations.

For now, let's deal with hurdle i, but drop the i for convenience, and drop the hat from $\hat{\beta}$. What Cox did was to write the link between λ_z, the failure probability for persons with value $Z=z$ and the failure probability λ_0 for persons with value $Z=0$ using the same logistic regression formulation he used for discrete time in section 6. Thus, if we take persons with a 'typical' or central value, say $\tilde{z}$, we can write their failure probabilty $\lambda_{\tilde{z}}$ in terms of 'two odds and an an odds ratio' (*sounds like a movie title !*), as

$$\frac{\lambda_{\tilde{z}}}{1-\lambda_{\tilde{z}}}=\exp\{\beta\tilde{z}\}\times\frac{\lambda_0}{1-\lambda_0}.$$

Cox uses π instead of $\lambda_{\tilde{z}}$, so his model is

$$\frac{\pi}{1-\pi}=\exp\{\beta\tilde{z}\}\times\frac{\lambda_0}{1-\lambda_0}.$$

Written in terms of λ_0 itself, it is

$$\lambda_0=\frac{odds_0}{1+odds_0}=\frac{\pi\exp\{-\beta\tilde{z}\}}{(1-\pi)+\pi\exp\{-\beta\tilde{z}\}}.$$

Now let us consider the (ML) estimation of π. We have as data a series of r individuals with covariate vectors $z_1,\ldots,z_j,\ldots,z_r$. The odds of failure for person j are

$$\exp\{\beta z_j\}\times\frac{\lambda_0}{1-\lambda_0}.$$

These odds can be rewitten, in reference to the odds involving the typical person, as

$$\exp\{\beta(z_j-\tilde{z})\}\times\frac{\pi}{1-\pi}.$$

Now $\exp\{\beta(z_j-\tilde{z})\}$ is the hazard-ratio for z_j relative to $\tilde{z}$. If we shorten it to e_j, the odds of failure for covariate z_j become

$$e_j\times\frac{\pi}{1-\pi},$$

and so the probability of failure becomes

$$\pi_j=\frac{odds_j}{odds_j+1}=\frac{\pi e_j}{(1-\pi)+\pi e_j}.$$

As data, we have a series of r independent but not identically distributed Bernoulli observations, governed by r probabilities, π_1 to π_r. But the set of r probabilities is held together by a single parameter π, and r observation-specific offsets involving the e's.

Denote the realizations of the r Bernoulli's as y_1 to y_r. If we now go through the ML estimation, we will find, as usual, that the single estimating equation is simply

$$\sum y_j=\sum E[y_j|\pi]=\sum\hat{\pi}_j,$$

where $\hat{\pi}_j$ is as defined above.

Since $\sum y_j=m$, we therefore have

$$m=\sum\frac{\pi e_j}{(1-\pi)+\pi e_j}\quad;\quad r-m=\sum\frac{1-\pi}{(1-\pi)+\pi e_j}.$$

Dividing the left identity by π and the right one by $1-\pi$, and subtracting the right from the left, we obtain

$$\frac{m}{\pi}-\frac{r-m}{1-\pi}=\frac{m-r\pi}{\pi(1-\pi)}=\sum\frac{e_j-1}{(1-\pi)+\pi e_j},$$

which leads directly to equation 37 in Cox's paper:

$$\hat{\pi}=\frac{m}{r}-\frac{\hat{\pi}(1-\hat{\pi})}{r}\sum\frac{e_j-1}{(1-\hat{\pi})+\hat{\pi}e_j}.$$

Comment:

The recasting of the (odds and thus) probability of failure for covariate z_j in terms of the odds (and thus) probability for an arbitrarily chosen value $\tilde{z}$ seems like an unnecessary step today, what with the computing power available. One reason to do so is to avoid large powers, especially if z has values far from zero, such as calendar year, cholesterol, etc.

The *key* step was to express the link between the probability λ_0 at each z value and the probability at $z = 0$ in terms of a baseline *odds* $\frac{\lambda_0}{1-\lambda_0}$, and an odds ratio $\exp\{\beta z\}$. Had he treated $\exp\{\beta z\}$ as a multiplier of λ_0, he could have created probabilities that exceeded 1. With the natural (canonical) link, he kept the probabilities within their permitted range.

If we took $\tilde{z}$ to be 0, the expression for $\hat{\pi}$ would still have been

$$\hat{\pi} = \frac{m}{r} - \frac{\hat{\pi}(1-\hat{\pi})}{r} \sum \frac{e_j - 1}{(1-\hat{\pi}) + \hat{\pi} e_j},$$

but now $e_j = \exp\{\beta z_j\}$ might be quite large. If the z's were already centered, the e_j's would fluctuate around 1.

Mathematica, numerical e.g.: $r = 3; m = 2; z = \{2, 4, 6\}, e^{\hat{\beta}} = 2$.

```
zTilde=Apply[Plus,z]/Length[z]  ; zTilde
b=Log[2.0]; eCentered = Exp[b (z-zTilde)];
Apply[Plus,eCentered]
y=Join[Table[1,{m}],Table[0,{r-m}]];
s = Apply[Plus, (eCentered-1)/( (1-p)+ p eCentered )]

zTilde= {2, 4, 6};  Sum[eCentered]= 4; s= 5.25
Apply[Plus, (eCentered-1)/( (1-p)+ p eCentered )]=
  -0.75        3.
---------- + --------
1 - 0.75 p   1 + 3. p

phat=(p/.NSolve[p == m/r - (p(1-p)/r) * s, p])[[2]]
lambda0hat = ( phat * Exp[-b zTilde] ) /
             ( phat * Exp[-b zTilde]  + (1-phat) )
U=(1/phat)Apply[Plus,1/( (1-phat)+ phat*Take[eCentered,m] ) ] -
(1/(1-phat))Apply[Plus,Drop[eCentered,m] /
                  ( (1-phat)+ phat*Drop[eCentered,m] ) ]
```

phat=0.711164 lambda0hat =0.133363 U = -8.88178 10^{-16}

Mathematica with symbolic example

```
eCentered = Table[e[j],{j,1,r}];
odds = eCentered * p/(1-p)
L = odds^y / (odds+1)
logL = Apply[Plus,D[Log[L],p]]
Simplify[logL]

Out[36]=
       1                  1                           e[3]
---------------- + ---------------- + -------------------------
    2    2              2    2         (-1 + p) (1 - p + p e[3])
p - p   + p  e[1]   p - p   + p  e[2]
```

This suggests that the estimating equation can also be written as

$$\frac{1}{\hat{\pi}} \sum_{1}^{m} \frac{1}{(1-\hat{\pi}) + \hat{\pi} e_j} = \frac{1}{1-\hat{\pi}} \sum_{m+1}^{r} \frac{e_j}{(1-\hat{\pi}) + \hat{\pi} e_j}.$$

With time and patience, this could be re-written in the same form as the estimating equation given above. I did check however that it is correct in some numerical examples – see the one (opposite) with $m = 2$ events in a riskset of size $r = 3$ and covariate values $z_1 = 2, z_2 = 4, z_3 = 6$, mean 4. Whereas 2/3 in the riskset failed, they have values of the covariate that far exceed zero. The estimated probability $\hat{\lambda}_{\tilde{z}}$ for persons with $z = \tilde{z} = \bar{z} = 4$ is 0.71. The estimated probability $\hat{\lambda_0}$ for persons with $z = 0$ is thus only 0.133 or 1/6.

Cox's method: equivalent to logistic regression [i.e.. binomial regression, logit link] with an offset !

```
y=c(1,1,0); z=c(2,4,6); beta.hat=log(2); o = log(exp(beta.hat*z))
logit.lambda0.hat = coef(glm(y~1,family=binomial,offset=o))
exp(logit.lambda0.hat)/(1+exp(logit.lambda0.hat) )
(Intercept)
  0.1333631   ---- doesn't change if use  y=c(0,1,1) or y=c(1,0,1)
```

Kalbfleisch & Prentice estimator, 1973 – verbatim from Biometrika

4. THE ESTIMATION OF THE SURVIVOR FUNCTION

Cox (1972) suggested an iterative procedure for estimating the survivor function at $z = \tilde{z}$ given a set of data from the model (1) and an estimate of β. In obtaining the estimate, it was assumed that $\lambda_0(t)$ was identically zero aside from mass points at the observed failure times. The data were taken as having arisen from his assumed logistic discrete analogue of the model (1) and a separate maximum likelihood estimation of the hazard at each failure point was then proposed. One difficulty with this approach is that the logistic model cannot be obtained by grouping the continuous model (1), as was pointed out in section 3. As a result of this, the estimates of the survivor function obtained for different values of z relate to one another through the logistic model only; continuous survivor functions cannot be constructed which are arbitrarily close to these estimates and relate to one another through the model (1). The result, therefore, is not a legitimate supremum of the likelihood defined on the model (1).

A procedure more compatible with the continuous model is obtained by considering the discrete model (6) instead of the logistic case. Following Cox's approach, the estimate $\hat{\beta}$ of β from the marginal likelihood is used. If the model (6) is adopted, the maximum likelihood estimate of the contribution $\hat{\lambda}_0(t_{(i)})dt_{(i)} = 1 - \hat{\alpha}_i$ to the hazard at $t = t_{(i)}$ is given by

$$\sum_{k\epsilon F_i} \frac{\exp(\hat{\beta} z_k)}{1 - \hat{\alpha}_i^{\exp(\hat{\beta} z_k)}} = \sum_{l\epsilon R(t_{(i)})} \exp(\hat{\beta} z_l), \qquad (7)$$

where F_i is the set of individuals failing at $t_{(i)}$. If only a single failure occurs at $t_{(i)}$ ($m_i = 1$), or all individuals in F_i have the same covariate value, this equation can be solved analytically for $\hat{\alpha}_i$. Otherwise, an iterative solution is required. A suitable starting value for the iteration is

$$\hat{\alpha}_{i0} = \exp\{-m_i / \sum_{l\epsilon R(t_{(i)})} \exp(\hat{\beta} z_l)\} \qquad (8)$$

In fact, since expression (8) is obtained by substituting

$$1 + \exp(\hat{\beta} z_k) \log \hat{\alpha}_i \approx \exp\{\exp(\hat{\beta} z_k) \log \hat{\alpha}_i\}$$

in (7), we expect $\hat{\alpha}_{i0}$ will approximate $\hat{\alpha}_i$ very closely if there are many distinct failure times, that is the $\hat{\alpha}_i$'s are near 1. It now easily follows from (6) that the estimated survivor function for a covariate value $z = \tilde{z}$ is

$$\hat{\mathscr{F}}_{\tilde{z}}(t) = \prod_{\{i|t_{(i)}<t\}} \hat{\alpha}_i^{\exp(\hat{\beta}\tilde{z})}. \qquad (9)$$

Expression (9) is a legitimate supremum of the likelihood function for the continuous case while, as noted above, the logistic result is not. A sequence of continuous survivor functions can be constructed statisfying the model (1) which converge to (9) for all values of 2. As could be expected, when the covariate z is the same for all individuals sampled, (9) reduces to the Kaplan-Meier product limit estimate. The estimate (9) of the survivor function will typically require an iterative solution when ties are present in the data, although the computations are fairly simple. Further, (9) is a step function estimate of the survivor function and in many instances a continuous estimate would be preferable, especially for suggesting a parametric form for $\lambda_0(t)$ or for communicating information to non-statisticians. In the remainder of this section, a continuous estimate ofthe survivor function is derived which has the further advantage ofbeing computationally simpler than (9).

We begin by approximating the ...

Comment:

Notice how Kalbfleisch and Prentice focused on the probability of survival, $\alpha_0 = 1 - \lambda_0$, rather than on that of failure, λ_0.

Their maximum likelihood estimate of α can be arrived at by writing the likelihood as a product of r Bernoulli likelihoods,

$$\prod_{k\epsilon F} (1 - \alpha_{z_k}) \prod_{j\epsilon R-F} \alpha_{z_j}$$

Now, because of the (short-term) proportional hazards, $\alpha_{z_j} = \alpha_0^{\exp(\hat{\beta} z_j)}$. Thus we can write the likelihood as

$$\prod_{k\epsilon F} (1 - \alpha_0^{\exp(\hat{\beta} z_k)}) \prod_{j\epsilon R-F} \alpha_0^{\exp(\hat{\beta} z_j)}$$

Taking the derivative with respect to α_0 of $d \log L$ and setting it to zero, will, after some re-arrangement, lead to estimating equation (7).

Kalbfleisch and Prentice's method: equivalent to binomial regression with cloglog link, and offset !

```
> require(survival)
>
> r=3; m=2; fail=c( rep(1,m),rep(0,r-m) );
> t=rep(1,r) ; survive=1-fail ; z=2*(1:r)
>
> summary(coxph(Surv(t,fail)~z,data=ds))

   coef exp(coef) se(coef)      z    p
z -0.49     0.613    0.496 -0.988 0.32

> # use beta-hats's to create offsets = log(exp[beta.hat * z])
>
> beta.hat = coef(coxph(Surv(t,fail)~z),data=ds) ;
> o = z * beta.hat ; o
[1] -0.9798727 -1.9597453 -2.9396180
>
> ds=data.frame(fail,t,z,e,o); ds
  fail t z          e          o
1    1 1 2 0.37535890 -0.9798727
2    1 1 4 0.14089430 -1.9597453
3    0 1 6 0.05288593 -2.9396180

> ds.zero=data.frame(z=rep(0,r)) ; ds.zero
  z
1 0
2 0
3 0
>
> fit <- coxph( Surv(t,fail)~z,data=ds )
> sfit=survfit(fit,newdata=ds.zero,type="kaplan-meier") #actually K-P
> sfit$surv[1]
```

[1] **3.929164e-05**

```
> # kalbfleisch and prentice -- binomial with cloglog link
>
> b0.hat=coef(glm(fail~1,
      family=binomial(link=cloglog),offset=o),data=ds) ; b0.hat

(Intercept)    2.316923     s0.hat = exp(-exp(b0.hat)) ; s0.hat
```

(Intercept) 3.929506e-05

LARGER EXAMPLE... $\hat{\beta} = 1.143$.

```
> died=c(1,0,1,1,0,1,1);t=c(2,4,6,8,10,12,14);
> z = c(1,1,0,1,0,1,0); o = 1.143*z ;
> dta = data.frame(died,t,z,o) ; dta
  died  t z     o
1    1  2 1 1.143
2    0  4 1 1.143
3    1  6 0 0.000
4    1  8 1 1.143
5    0 10 0 0.000
6    1 12 1 1.143
7    1 14 0 0.000
> z=rep(0,7); d.zero=data.frame(z);
> fit = coxph( Surv(t,died)~z,data=dta)
> srv = survfit( fit, newdata=d.zero, type="kaplan-meier" )
> srv$surv[,1]          #type is actually kalbfleisch-prentice

[1] 0.931 0.830 0.713 0.454 0.000

n=length(died) ; s = 1; s0.hat=c(s) ; time=c(0)
for (i in 1:n){
 if (died[i]==1)
 { fail = ( t[i] == t[i:n] );
   time=c(time,t[i])
  b.0=coef(glm(fail~1,family=binomial(link=cloglog),offset=o[i:n]));
  S = exp( -exp(b.0) ); s = s * S; s0.hat=c(s0.hat,s)  } }
  s0.hat = round( s0.hat ,3) ; s0.hat
      1.000     0.931     0.830     0.713     0.454     0.000
```

This was put together hastily and may contain errors; corrections welcome.