An Example of Survival Analysis in R – using the recidivism data

1. Introduction to the recidivism data

• **week:** week of first arrest after release, or censoring time.
• **arrest:** the event indicator, equal to 1 for those arrested during the period of the study and 0 for those who were not arrested.
• **fin:** a dummy variable, equal to 1 if the individual received financial aid after release from prison, and 0 if he did not; financial aid was a randomly assigned factor manipulated by the researchers.
• **age:** in years at the time of release.
• **race:** a dummy variable coded 1 for blacks and 0 for others.
• **wexp**: a dummy variable coded 1 if the individual had full-time work experience prior to incarceration and 0 if he did not.
• **mar:** a dummy variable coded 1 if the individual was married at the time of release and 0 if he was not.
• **paro:** a dummy variable coded 1 if the individual was released on parole and 0 if he was not.
• **prio:** number of prior convictions.
• **educ:** education, a categorical variable, with codes 2 (grade 6 or less), 3 (grades 6 through 9), 4 (grades 10 and 11), 5 (grade 12), or 6 (some post-secondary).
• **emp1 — emp52:** dummy variables coded 1 if the individual was employed in the corresponding week of the study and 0 otherwise.

2. Load the "recid.csv" data to R and get ready to survival analysis

```
> dat = read.csv("recid.csv")
> head(dat)
  week arrest fin age race wexp mar paro prio educ emp1 emp2 …  emp52
1   20      1   0  27    1    0   0    1    3    3    0    0    0     …    NA
2   17      1   0  18    1    0   0    1    8    4    0    0    0     …    NA
3   25      1   0  19    0    1   0    1   13    3    0    0    0     …    NA
4   52      0   1  23    1    1   1    1    1    5    0    0    0     …     1
5   52      0   0  19    0    1   0    1    3    3    0    0    0     …     0
6   52      0   0  24    1    1   0    0    2    4    0    0    0     …     0

> attach(dat)
```

The package "survival" is used in each example in this lab.

```
> library(survival)
```

Before complex functions may be performed, the data has to be put into the proper format: **a survival object**. The function Surv() will be used to construct these survival objects for –

1) right-censored data: Surv(time, event)
2) left-truncated and right-censored data: Surv(time, time2, event)

Here we have the right-censored survival time "week" with the event indicator "arrest":

```
> mysurv = Surv(week, arrest)
> mysurv
[1] 20  17  25  52+ 52+ 52+ 23  52+ 52+ 52+ 52+ 52+ 37  52+ 25  46
 …
[419] 36  52+ 52+  8  15  52+ 19  52+ 12  52+ 52+ 52+ 52+ 52+
```

3. Obtain Kaplan-Meier estimate (by groups)

The function survfit(formula, …) is used to find the Kaplan-Meier estimate of the survival function.

```
> myKM1 = survfit(mysurv ~ 1)
> myKM1
Call: survfit(formula = mysurv ~ 1)

records   n.max n.start  events  median 0.95LCL 0.95UCL
    432     432     432     114      NA      NA      NA
```

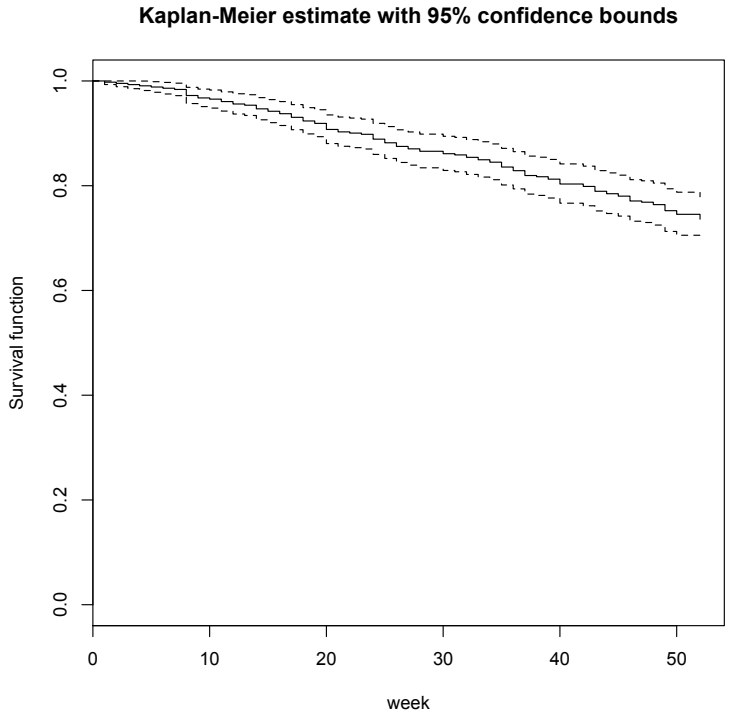
Why the "median failure time" is not available? Check the survival estimates:

```
> summary(myKM1)
Call: survfit(formula = mysurv ~ 1)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1    432       1    0.998 0.00231       0.993       1.000
    2    431       1    0.995 0.00327       0.989       1.000
    3    430       1    0.993 0.00400       0.985       1.000
   ……
   49    330       5    0.752 0.02077       0.713       0.794
```

```
  50    325       3    0.745 0.02096       0.705       0.788
  52    322       4    0.736 0.02121       0.696       0.779
```

Moreover, we can plot the K-M estimates with the 95% CIs:

```
> plot(myKM1, main=("Kaplan-Meier estimate with 95% confidence
bounds"), xlab = "week", ylab = "Survival function")
```



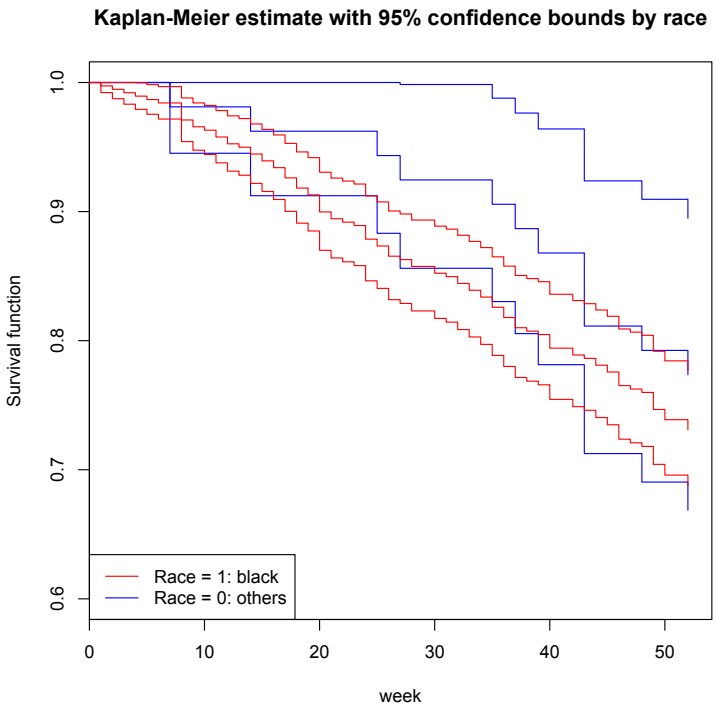
Taking advantage of the "formula" in survfit(), we can obtain the K-M estimates by race groups:

```
> myKM2 = survfit(mysurv ~ race)
> summary(myKM2)

Call: survfit(formula = mysurv ~ race)

                race=0
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    7     53       1    0.981  0.0187       0.945       1.000
   14     52       1    0.962  0.0262       0.912       1.000
  ……
   48     43       1    0.792  0.0557       0.690       0.910
   52     42       1    0.774  0.0575       0.669       0.895
```

```
             race=1
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1    379       1    0.997 0.00264       0.992       1.000
    2    378       1    0.995 0.00372       0.987       1.000
   ……
   50    283       3    0.739 0.02257       0.696       0.784
   52    280       3    0.731 0.02278       0.688       0.777
```

Then we can also plot the K-M estimates by groups:

```
> plot(myKM2, conf.int = T, col = c("blue", "red"),
main=("Kaplan-Meier estimate with 95% confidence bounds by race"),
xlab = "week", ylab = "Survival function", ylim = c(0.6,1))
> legend("bottomleft", c("Race = 1: black","Race = 0: others"), col
= c("red", "blue"), lty = 1)
```



The 95% CI bands for the two race groups overlap. Thus, the estimated higher survival function for race = 0 over that for race = 1 is not significant. We can confirm this by conducting (log-rank) test for the two groups.

To check the null hypothesis of no difference between the survival times, use survdiff(formula, rho = 0). The default is rho=0, which corresponds to the log-rank test. When rho=1, this is the "Peto & Peto modification of the Gehan-Wilcoxon test":

```
> survdiff(mysurv ~ race, rho = 0)
Call:
survdiff(formula = mysurv ~ race, rho = 0)

        N Observed Expected (O-E)^2/E (O-E)^2/V
race=0  53       12     14.7    0.4990     0.576
race=1 379      102     99.3    0.0739     0.576

 Chisq= 0.6  on 1 degrees of freedom, p= 0.448
```

4. Cox Proportional Hazards Model (PHM) – time independent covariates

The function coxph(formula, …) fits a Cox PH model to the supplied data. "formula" will be almost identical to fitting a linear model (via lm()) except that the response variable will be a survival object instead of a vector.

```
> myfit1 = coxph(mysurv ~ fin + age + race + wexp + mar + paro + prio,
+    ties = "breslow")
```

Here the parameter `ties=c("efron","breslow","exact")`. If there are no tied failure times all the three methods are equivalent.

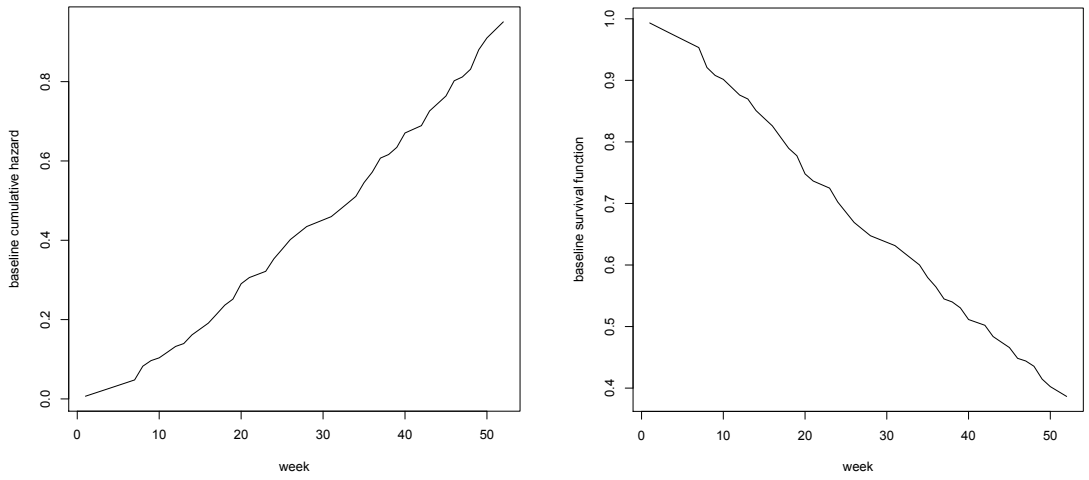
1) Nearly all Cox regression programs (e.g. STATA) use the Breslow method by default, but not this one.
2) The **Efron approximation** is used as the **default** here, it is more **accurate** when dealing with tied death times, and is as **efficient** computationally.
3) The "exact partial likelihood" is equivalent to a conditional logistic model, and is appropriate when the times are a small set of discrete values. If there are a large number of ties and (start, stop) style survival data the computational time will be excessive.

```
> summary(myfit1)

Call:
coxph(formula = mysurv ~ fin + age + race + wexp + mar + paro +
    prio, ties = "breslow")
```

```
  n= 432, number of events= 114

       coef exp(coef) se(coef)     z Pr(>|z|)
fin  -0.37902   0.68453  0.19136 -1.981  0.04763 *
age  -0.05725   0.94436  0.02198 -2.604  0.00921 **
race  0.31413   1.36907  0.30802  1.020  0.30780
wexp -0.15111   0.85975  0.21212 -0.712  0.47622
mar  -0.43278   0.64870  0.38179 -1.134  0.25698
paro -0.08498   0.91853  0.19575 -0.434  0.66418
prio  0.09111   1.09539  0.02863  3.182  0.00146 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

    exp(coef) exp(-coef) lower .95 upper .95
fin    0.6845     1.4609    0.4704    0.9961
age    0.9444     1.0589    0.9045    0.9859
race   1.3691     0.7304    0.7486    2.5039
wexp   0.8597     1.1631    0.5673    1.3030
mar    0.6487     1.5415    0.3069    1.3710
paro   0.9185     1.0887    0.6259    1.3481
prio   1.0954     0.9129    1.0356    1.1586

Concordance= 0.64  (se = 0.027 )
Rsquare= 0.074   (max possible= 0.956 )
Likelihood ratio test= 33.13  on 7 df,   p=2.509e-05
Wald test            = 31.98  on 7 df,   p=4.095e-05
Score (logrank) test = 33.38  on 7 df,   p=2.246e-05
```

As a by-product of Cox PHM, how to obtain the baseline (cumulative) hazard and corresponding baseline survival function?

```
> bch = basehaz(myfit1, centered = F)
> plot(bch$time, bch$hazard, type = "l",
+   xlab = "week", ylab = "baseline cumulative hazard")
> plot(bch$time, exp(-bch$hazard), type = "l",
+   xlab = "week", ylab = "baseline survival function")
```



5. **Cox Proportional Hazards Model (PHM) – time dependent covariates**

Time independent covariates are easy to work with in R, however, working with time-dependent covariates is an exercise in organization. There is a way to work around the fact that there aren't functions in R that will directly accept time-dependent covariates: use left-truncation liberally. This works on the basic principle that if there is one right-censored observation, say 45+, it is the same as having two observations that are left-truncated right-censored, (0, 12+) and (12, 45+), where the choice of splitting the observation at 12 was arbitrary. Usually we use the time points at which the covariates were observed as the cut-points. This is the workaround used in R to make time-dependent variables.

Therefore, we have to re-arrange the data from a wide format into a **long format**: repeated observations for each subject, and one row for each observation. Observations are **clustered** by the subject id. Moreover, to declare the data to be a left-truncated and right-censored dataset, we need to specify a **start** time and **end** time for each observation interval. Here we use the week 1 through 52 (time points at which the covariate "emp" is observed) as the cut-points. Then for each observation interval, the end time would be the current week number, and the start time would be the previous week.

You can use whatever way you like to manage the dataset in R. Here as an example, I created a new data.frame named "newdat" (see related R code in Rlab_recid.R).

```
> head(newdat)
  id week arrest fin age race wexp mar paro prio educ emp start end
1  1   20      1   0  27    1    0    0    1    3    3    0     0   1
2  1   20      1   0  27    1    0    0    1    3    3    0     1   2
3  1   20      1   0  27    1    0    0    1    3    3    0     2   3
4  1   20      1   0  27    1    0    0    1    3    3    0     3   4
5  1   20      1   0  27    1    0    0    1    3    3    0     4   5
6  1   20      1   0  27    1    0    0    1    3    3    0     5   6
```

Every person has 52 records because there are 52 emp variables (emp1-emp52). So we keep the records no later than the failure/censored week.

```
> newdat = newdat[newdat$end <= newdat$week,]
```

Arrest is the status of at the end of the follow-up time. So we change it to 0 for previous weeks.

```
> newdat$arrest[newdat$end < newdat$week] = 0
```

Finally, the Cox model looked like:

```
> coxph(Surv(start, end, arrest) ~ cluster(id) + fin + age + race +
+    wexp + mar + paro + prio + emp, data = newdat, ties = "breslow")

Call:
coxph(formula = Surv(start, end, arrest) ~ cluster(id) + fin +
    age + race + wexp + mar + paro + prio + emp, data = newdat,
    ties = "breslow")


       coef exp(coef) se(coef) robust se     z      p
fin  -0.3560    0.700   0.1911    0.195 -1.825 6.8e-02
age  -0.0461    0.955   0.0217    0.025 -1.845 6.5e-02
race  0.3386    1.403   0.3096    0.296  1.143 2.5e-01
wexp -0.0275    0.973   0.2113    0.218 -0.126 9.0e-01
mar  -0.2929    0.746   0.3829    0.396 -0.740 4.6e-01
paro -0.0644    0.938   0.1947    0.198 -0.325 7.5e-01
prio  0.0847    1.088   0.0289    0.029  2.918 3.5e-03
emp  -1.3246    0.266   0.2507    0.251 -5.277 1.3e-07

Likelihood ratio test=68.3  on 8 df, p=1.08e-11  n= 19809, number of events=
114
```

For the time-independent variables, the coefficients and test statistics are pretty much the same as the previous model, however the time-dependent variable EMP has the strongest effect. But we notice that sometimes arrests affect employment status rather than vice versa. If someone is arrested in a week, the probability of working full time during that week is very small. This potential **reverse causation** is a common problem in time dependent variables. Let's modify the model by using the employment status in previous week instead of the current week.

```
> ### using the "employment status for previous week" as the covariate
> newdat$empprev = c(NA, newdat$emp[-nrow(newdat)])
> for (i in 1:n){
+   ### first observation of empprev for each subject is NA
+   newdat$empprev[newdat$id == i][1] = NA
+ }
```

Then use "empprev" instead of "emp":

```
> coxph(Surv(start, end, arrest) ~ cluster(id) + fin + age + race +
+   wexp + mar + paro + prio + empprev, data = newdat, ties = "breslow")

Call:
coxph(formula = Surv(start, end, arrest) ~ cluster(id) + fin +
    age + race + wexp + mar + paro + prio + empprev, data = newdat,
    ties = "breslow")


          coef exp(coef) se(coef) robust se     z     p
fin    -0.3506    0.704   0.1918    0.1947 -1.801 0.0720
age    -0.0496    0.952   0.0219    0.0251 -1.979 0.0480
race    0.3216    1.379   0.3091    0.2931  1.097 0.2700
wexp   -0.0491    0.952   0.2131    0.2173 -0.226 0.8200
mar    -0.3445    0.709   0.3831    0.3903 -0.883 0.3800
paro   -0.0474    0.954   0.1963    0.1977 -0.240 0.8100
prio    0.0916    1.096   0.0288    0.0283  3.233 0.0012
empprev -0.7838    0.457   0.2181    0.2166 -3.619 0.0003

Likelihood ratio test=46.9  on 8 df, p=1.6e-07  n= 19377, number of events=
113
  (432 observations deleted due to missingness)
```