

# DEM 7223 - Event History Analysis - Comparing Survival Times Between Groups

Corey S. Sparks, PhD

## Product Limit Estimation

[Kaplan and Meier \(1958\)](#) derived an estimator of the survivorship function for a sample of censored and uncensored cases.

- The method figured survival to a time,  $t$ , is the product of the survival from all previous time points.
- i.e. you can only get to time 3, if you survive time 1 and time 2, etc

We can write this as

$$\hat{S}(t) = (1 - p(\hat{t}_1))(1 - p(\hat{t}_2)) \dots (1 - p(\hat{t}_j))$$

Unlike the life-table and discrete time methods of estimating survival, which lumped time into discrete periods, K-M uses the information contained in the actual duration.

Each K-M interval begins with a single event time, and ends just prior to the next event time.

## Kaplan-Meier Estimation

The K-M estimator is written:

$$\hat{S}(t) = \prod_{t_i \leq t} \frac{n_i - d_i}{n_i} = \prod_{t_i \leq t} \left[ 1 - \frac{d_i}{Y_i} \right]$$

Where the  $t_i$  are the ranked survival times,  $n_i$  is the number of individuals at risk at each time,  $t_i$  is time,  $d_i$  is the number of events at each time.

When censoring is present, you define  $n_i$  by subtracting out the number of censored cases at that particular time.

```

t1<-data.frame(Time_Interval=c("0,1)", "[1,2)", "[2,4)", "[4, 5)", "[5, 6)", "[6+)"),
               time=c(0, 1, 2, 4, 5, 6),
               n=c(100,NA, NA, NA, NA, NA),
               d=c(15, 15, 11, 0, 0, 2),
               c=c(0,2,5,2,0,2),
               prob=c(1,NA, NA, NA, NA, NA),
               St=c(1,NA, NA, NA, NA, NA))

for (i in 2:6){
  t1$n[i]<-t1$n[i-1]- t1$d[i-1] -t1$c[i-1]
  t1$prob[i]<-1-(t1$d[i]/t1$n[i])
}

t1$St<-cumprod(t1$prob)

t1%>%
  kable(format ="simple" )#%>%

```

Time_Interval	time	n	d	c	prob	St
[0,1)	0	100	15	0	1.0000000	1.0000000
[1,2)	1	85	15	2	0.8235294	0.8235294
[2,4)	2	68	11	5	0.8382353	0.6903114
[4, 5)	4	52	0	2	1.0000000	0.6903114
[5, 6)	5	50	0	0	1.0000000	0.6903114
[6+)	6	50	2	2	0.9600000	0.6626990

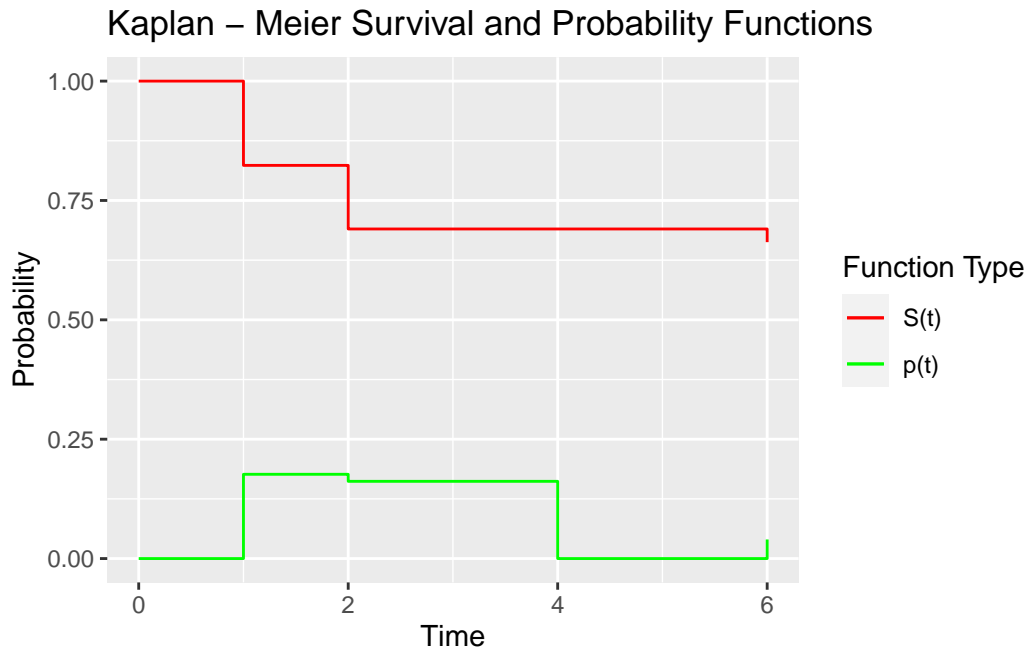
```

# column_spec(1:4, border_left = T, border_right = T)#%>%
# kable_styling()

t1%>%
  ggplot()+
  geom_step(aes(x=time, y=St,color="a"))+
  geom_step(aes(x=time, y=1-prob, color="b"))+
  scale_colour_identity(guide="legend")+
  scale_colour_manual(name = 'Function Type',
                      values =c('a'='red','b'='green'),
                      labels = c('S(t)','p(t)'))+
  xlab("Time")+ylab("Probability")+
  ggtitle("Kaplan - Meier Survival and Probability Functions")

```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



### K-M and the hazard function

The exact estimate of the K-M hazard function actually depends on the width of the time interval at each observed time point.

$$h(\hat{t}_j) = \frac{p_{KM}(\hat{t}_j)}{\text{width}_j}$$

and you can get this estimate from the `muhaz` library in R.

### Variance in the K-M Estimates

Since the K-M survival function is a statistical estimate, it also has uncertainty to it. To measure the uncertainty, or variance in the estimate, the traditional method is to use the Greenwood formula.

$$\text{Var}(S(t)) = \hat{S}(t)^2 \sum_{t_i \leq t} \frac{d_i}{n_i(n_i - d_i)}$$

and standard error equal to

$$s.e.(S(t)) = \sqrt{\text{Var}(S(t))}$$

If we have a standard error of the survival function, the assuming the sampling distribution of the survival function is normal, we can calculate a normal confidence interval for the survival function at each time:

$$c.i.(S(t)) = \hat{S}(t) \pm z_{1-\alpha/2} * s.e.(S(t))$$

where  $z$  is the standard normal variate corresponding to the the  $1 - \alpha/2$  level of confidence.

### **Estimating the cumulative hazard function.**

If we have estimates of  $\hat{h}(t_j)$ , we can either calculate the value of the cumulative hazard function using the relationship among survival function:

$$H(t) = -\log S(t)$$

or use the Nelson-Aalen estimator:

$$\hat{H}(t) = \sum_{t_i \leq t} \frac{d_i}{Y_i}$$

### **Comparing Survival curves between groups**

Often in our data there are distinct groups for which we want to compare survival patterns  
\* e.g. treatment vs. control group in clinical trial \* e.g. proportion of women without a first birth by education category

To do this we typically construct a variable that represents an identifier for members of each group. This can be referred to as an indicator, or class, variable.

This is the same process as doing a two sample test for a regular outcome, such as a t-test or chi square test.

- One limitation of survival data is that they are typically skewed, meaning their distribution is not symmetrical
- This means that many traditional hypothesis test (t-test, z-test) for comparison of central tendency are no appropriate
- Instead we use a variety of non-parametric methods

- Simply meaning that these tests are not dependent on the shape of the distribution or on the parameters of the distribution
- Also, due to *censoring*, traditional distributional parameters like the mean are less meaningful, so tests on said parameters would be incorrect

## Graphical methods of comparison

The first step on our examination of between group comparison is the *inter-ocular traumatic test*

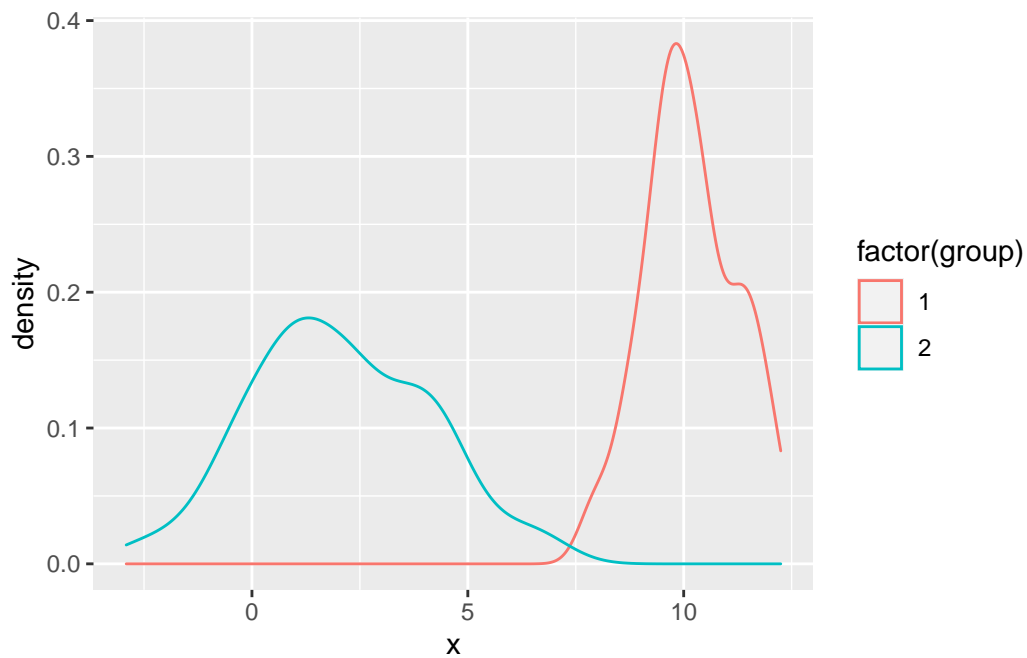
You may not be familiar with this test, but in general, if you look at a plot, and if you think there's a difference, say between two lines, there usually is, and many times the human eye is a more discerning test than anything.

Warning: ``data_frame()`` was deprecated in tibble 1.1.0.

Please use ``tibble()`` instead.

This warning is displayed once every 8 hours.

Call ``lifecycle::last_lifecycle_warnings()`` to see where this warning was generated.



So, plot the survival curves, with confidence intervals, your eye can usually detect if there is a difference

Under traditional statistics thinking, if the confidence intervals of the two curves overlap for their entire lengths, then the two groups are equivalent, if the confidence interval for the curves do not overlap at ANY point along the curve, they are different, simple, no?

The *statistical* way of doing this, beyond looking at things, is the realm of Mantel-Haenszel test. R implements this test for 2 or  $k$  groups using the `survdiff()` function. It uses the method of [Harrington and Fleming \(1982\)](#) which can weight the difference in survival curves flexibly, giving more or less weight to earlier or later survival times.

The classic Mantel-Haenszel test is just a  $\chi^2$  test for independence.

At each time point,  $t_i$ , consider the following table for 2 groups:

Group	1	0	total
Fail	$d_{1i}$	$d_{0i}$	$d_i$
Do not fail	$n_{1i} - d_{1i}$	$n_{0i} - d_{0i}$	$n_i - d_i$
At risk	$n_{1i}$	$n_{0i}$	$n_i$

Figure 1: M-H Test

If we sum the differences between the observed and expected failures across all time points, we arrive with:

Using 1 group as the basis:

$$e_{i1} = \frac{n_{i1}d_{i1}}{n_i}$$

is the expected number of failures. The general form of the test is then:

$$Q = \frac{\sum_i w_i (d_{i1} - e_{i1})^2}{\sum_i w_i v_{i1}}$$

Where  $v_{i1}$  is the variance in the number of events in group 1. This test follows a  $\chi^2$  distribution with 1 degree of freedom.

The value of  $w_i$  allows great flexibility for these tests, and is called the weight function at time  $i$ . This allows the analyst to specify how much you want to weight the difference in survival at a particular time point.

This testing logic also extends to  $k$ -groups, so instead of doing an ANOVA test, you would do the  $k$ -group test following this method.

## Example

This example will illustrate how to test for differences between survival functions estimated by the Kaplan-Meier product limit estimator. The tests all follow the methods described by Harrington and Fleming (1982) [Link](#).

The first example will use as its outcome variable, the event of a child dying before age 1. The data for this example come from the data [Demographic and Health Survey for 2012](#) children's recode file. This file contains information for all births in the last 5 years prior to the survey.

The second example, we will examine how to calculate the survival function for a longitudinally collected data set. Here I use data from the [ECLS-K](#). Specifically, we will examine the transition into poverty between kindergarten and fifth grade.

```
#load libraries
library(haven)
library(survival)
library(car)
```

Loading required package: carData

Attaching package: 'car'

The following object is masked from 'package:dplyr':

recode

The following object is masked from 'package:purrr':

some

```
library(muhaz)

dat<-read_dta("../data/ZAKR71FL.DTA")
dat<-zap_labels(dat)
```

## Event - Infant Mortality

In the DHS, they record if a child is dead or alive and the age at death if the child is dead. This can be understood using a series of variables about each child.

If the child is alive at the time of interview, then the variable B5==1, and the age at death is censored.

If the age at death is censored, then the age at the date of interview (censored age at death) is the date of the interview - date of birth (in months).

If the child is dead at the time of interview, then the variable B5!=1, then the age at death in months is the variable B7. Here we code this:

```
dat$death.age<-ifelse(dat$b5==1,
                      (((dat$v008))-(((dat$b3))))
                      ,dat$b7)

#censoring indicator for death by age 1, in months (12 months)
dat$d.event<-ifelse(is.na(dat$b7)==T|dat$b7>12,0,1)
dat$d.eventfac<-factor(dat$d.event)
levels(dat$d.eventfac)<-c("Alive at 1", "Dead by 1")
table(dat$d.eventfac)
```

Alive at 1	Dead by 1
3419	129

## Comparing Two Groups

We will now test for differences in survival by characteristics of the household. First we will examine whether the survival chances are the same for children in relatively high ses (in material terms) households, compared to those in relatively low-ses households.

This is the equivalent of doing a t-test, or Mann-Whitney U test for differences between two groups.

```
dat$highses<-Recode(dat$v190,
                    recodes ="1:3 = 0; 4:5=1; else=NA")

fit <-survfit(Surv(death.age, d.event)~highses, data=dat)
```



```
fit%>%
  ggsurvfit() +
  labs(title = "Infant mortality survival plot",
        subtitle = "By household SES",
        x = "Years", y = "Survival Probability")+
  xlim(c(0, 12))+
  ylim(c(.95, 1))
```

Warning: Removed 96 row(s) containing missing values (geom\_path).



```
summary(fit)
```

Call: `survfit(formula = Surv(death.age, d.event) ~ highses, data = dat)`

	highses=0						
time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI	
0	2557	71	0.972	0.00325	0.966	0.979	
1	2470	3	0.971	0.00332	0.965	0.978	
2	2415	6	0.969	0.00345	0.962	0.975	
3	2360	6	0.966	0.00359	0.959	0.973	

4	2316	7	0.963	0.00374	0.956	0.971
5	2264	2	0.962	0.00379	0.955	0.970
6	2219	1	0.962	0.00381	0.955	0.969
7	2180	1	0.962	0.00383	0.954	0.969
8	2146	4	0.960	0.00393	0.952	0.967
9	2111	3	0.958	0.00400	0.951	0.966
10	2070	2	0.957	0.00405	0.950	0.965
11	2033	1	0.957	0.00408	0.949	0.965
12	1991	5	0.955	0.00420	0.946	0.963

highses=1						
time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
0	991	11	0.989	0.00333	0.982	0.995
3	941	2	0.987	0.00364	0.980	0.994
4	918	3	0.984	0.00407	0.976	0.992
12	779	1	0.982	0.00426	0.974	0.991

Gives us the basic survival plot.

Next we will use `survdif()` to test for differences between the two or more groups. The `survdif()` function performs the log-rank test to compare the survival patterns of two or more groups.

```
#two group compairison
survdif(Surv(death.age, d.event)~highses, data=dat)
```

Call:

```
survdif(formula = Surv(death.age, d.event) ~ highses, data = dat)
```

	N	Observed	Expected	(O-E) <sup>2</sup> /E	(O-E) <sup>2</sup> /V
highses=0	2557	112	92.8	3.97	14.4
highses=1	991	17	36.2	10.18	14.4

Chisq= 14.4 on 1 degrees of freedom, p= 2e-04

In this case, we see no difference in survival status based on household SES.

How about rural vs urban residence?

```
dat<-dat%>%
  mutate(rural = car::Recode(v025,
                             recodes ="2 = '0rural'; 1='1urban'; else=NA",
```

```

as.factor = T))

fit2<-survfit(Surv(death.age, d.event) ~ rural,
              data=dat,
              conf.type = "log")

summary(fit2)

```

```

Call: survfit(formula = Surv(death.age, d.event) ~ rural, data = dat,
              conf.type = "log")

```

```

              rural=0rural
time n.risk n.event survival std.err lower 95% CI upper 95% CI
  0   1685     41    0.976 0.00375    0.968    0.983
  1   1631      2    0.974 0.00384    0.967    0.982
  2   1597      5    0.971 0.00407    0.963    0.979
  3   1558      3    0.970 0.00420    0.961    0.978
  4   1534      3    0.968 0.00433    0.959    0.976
  8   1418      4    0.965 0.00453    0.956    0.974
  9   1386      2    0.964 0.00463    0.955    0.973
 10   1363      2    0.962 0.00473    0.953    0.971
 12   1312      4    0.959 0.00494    0.950    0.969

```

```

              rural=1urban
time n.risk n.event survival std.err lower 95% CI upper 95% CI
  0   1863     41    0.978 0.00340    0.971    0.985
  1   1816      1    0.977 0.00344    0.971    0.984
  2   1774      1    0.977 0.00348    0.970    0.984
  3   1743      5    0.974 0.00369    0.967    0.981
  4   1700      7    0.970 0.00397    0.962    0.978
  5   1664      2    0.969 0.00405    0.961    0.977
  6   1636      1    0.968 0.00409    0.960    0.976
  7   1599      1    0.968 0.00414    0.960    0.976
  9   1539      1    0.967 0.00418    0.959    0.975
 11   1493      1    0.966 0.00423    0.958    0.975
 12   1458      2    0.965 0.00433    0.957    0.974

```

```

fit2 %>%
  ggsvfit(xlim=c(0,12),
          ylim=c(.95, 1),

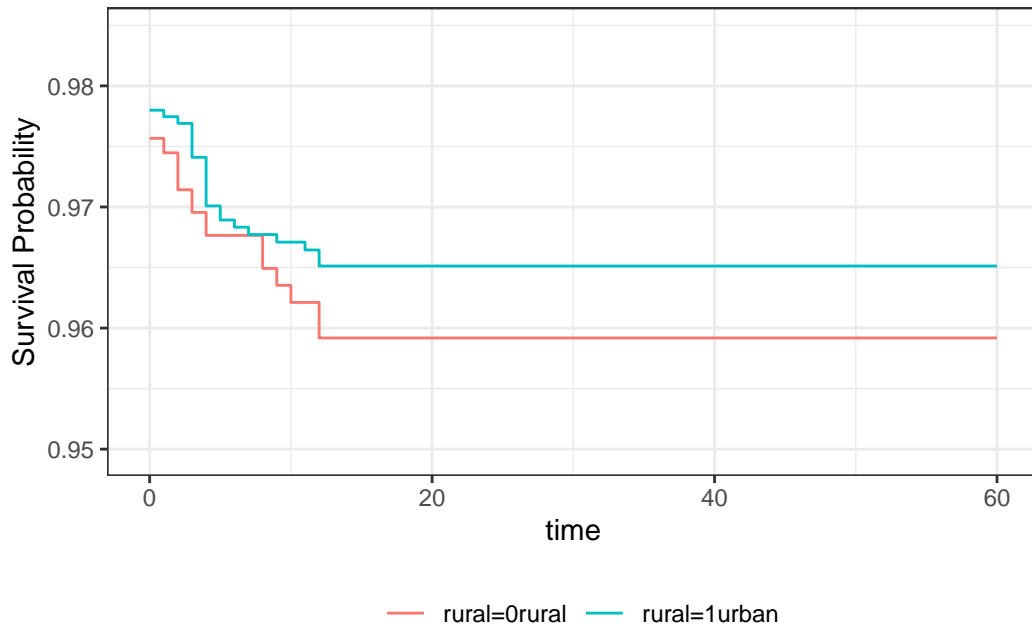
```

```

conf.int=T,
title="Survival Function for Infant mortality - Rural vs Urban Residence")

```

Warning: Ignoring unknown parameters: xlim, ylim, conf.int, title



## Two- sample test

```

survdif(Surv(death.age, d.event)~rural, data=dat)

```

Call:

```

survdif(formula = Surv(death.age, d.event) ~ rural, data = dat)

```

	N	Observed	Expected	$(O-E)^2/E$	$(O-E)^2/V$
rural=0rural	1685	66	61.2	0.376	0.727
rural=1urban	1863	63	67.8	0.340	0.727

Chisq= 0.7 on 1 degrees of freedom, p= 0.4

```
prop.table(table(dat$d.event, dat$rural), margin = 2)
```

```
      0rural      1urban
0 0.96083086 0.96618357
1 0.03916914 0.03381643
```

```
chisq.test(table(dat$d.event, dat$rural))
```

Pearson's Chi-squared test with Yates' continuity correction

```
data:  table(dat$d.event, dat$rural)
X-squared = 0.57882, df = 1, p-value = 0.4468
```

Which shows a statistically significant difference in survival between rural and urban children, with rural children showing lower survivorship at all ages.

We can also compare the 95% survival point for rural and urban residents

```
quantile(fit2, probs=.05)
```

```
$quantile
```

```
      5
rural=0rural NA
rural=1urban NA
```

```
$lower
```

```
      5
rural=0rural 12
rural=1urban NA
```

```
$upper
```

```
      5
rural=0rural NA
rural=1urban NA
```

## k - sample test

Next we illustrate a k-sample test. This would be the equivalent of the ANOVA if we were doing ordinary linear models.

In this example, I use the v024 variable, which corresponds to the region of residence in this data. Effectively we are testing for differences in risk of infant mortality by region.

```
table(dat$v024, dat$d.eventfac)
```

	Alive at 1	Dead by 1
1	202	4
2	432	18
3	278	8
4	304	14
5	538	17
6	380	15
7	358	12
8	472	29
9	455	12

```
survdif(Surv(death.age, d.event)~v024, data=dat)
```

Call:

```
survdif(formula = Surv(death.age, d.event) ~ v024, data = dat)
```

	N	Observed	Expected	(O-E) <sup>2</sup> /E	(O-E) <sup>2</sup> /V
v024=1	206	4	7.55	1.6728	1.8038
v024=2	450	18	16.37	0.1617	0.1881
v024=3	286	8	10.35	0.5348	0.5904
v024=4	318	14	11.50	0.5430	0.6053
v024=5	555	17	20.14	0.4888	0.5881
v024=6	395	15	14.47	0.0193	0.0221
v024=7	370	12	13.50	0.1677	0.1902
v024=8	501	29	18.17	6.4612	7.6357
v024=9	467	12	16.94	1.4402	1.6833

Chisq= 11.7 on 8 degrees of freedom, p= 0.2

Which does not show significant variation in survival between regions.

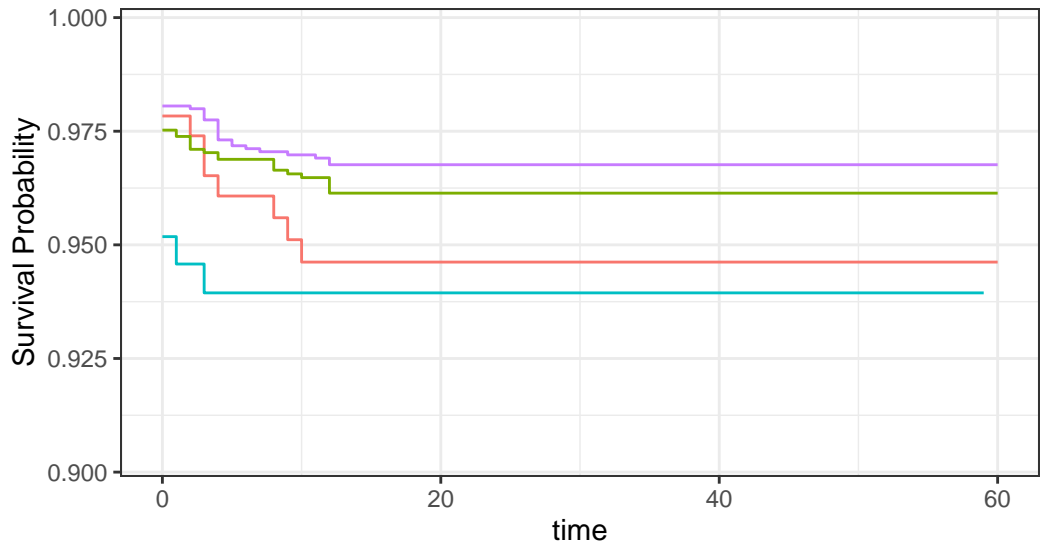
Lastly, we examine comparing survival across multiple variables, in this case the education of the mother (**secedu**) and the rural/urban residence **rural**:

```
dat<-dat%>%  
  mutate(secedu=Recode(v106, recodes ="2:3 = 1; 0:1=0; else=NA"))  
  
table(dat$secedu, dat$d.eventfac)
```

	Alive at 1	Dead by 1
0	375	22
1	3044	107

```
fit4<-survfit(Surv(death.age, d.event)~rural+secedu, data=dat)  
  
#summary(fit4)  
  
fit4 %>%  
  ggsvfit(conf.int = F,  
          risk.table = F,  
          title = "Survivorship Function for Infant Mortality",  
          xlab = "Time in Months",  
          xlim = c(0,12),  
          ylim=c(.9, 1))
```

Warning: Ignoring unknown parameters: conf.int, risk.table, title, xlab, xlim, ylim



rural=0rural, secedu=0    rural=0rural, secedu=1    rural=1urban, secedu=0    rural=1urban, secedu=1

```
# test
survdif(Surv(death.age, d.event)~rural+secedu, data=dat)
```

Call:

```
survdif(formula = Surv(death.age, d.event) ~ rural + secedu,
  data = dat)
```

	N	Observed	Expected	(O-E) <sup>2</sup> /E	(O-E) <sup>2</sup> /V
rural=0rural, secedu=0	231	12	8.47	1.4688	1.5960
rural=0rural, secedu=1	1454	54	52.73	0.0306	0.0526
rural=1urban, secedu=0	166	10	5.98	2.7024	2.8776
rural=1urban, secedu=1	1697	53	61.82	1.2579	2.4525

Chisq= 5.5 on 3 degrees of freedom, p= 0.1

Which shows a marginally significant difference between at *least* two of the groups, in this case, I would say that it's most likely finding differences between the Urban, low Education and the Rural high education, because there have the higher ratio of observed vs expected.



## Survival analysis using survey design

This example will cover the use of R functions for analyzing complex survey data. Most social and health surveys are not simple random samples of the population, but instead consist of respondents from a complex survey design. These designs often stratify the population based on one or more characteristics, including geography, race, age, etc. In addition the designs can be multi-stage, meaning that initial strata are created, then respondents are sampled from smaller units within those strata. An example would be if a school district was chosen as a sample strata, and then schools were then chosen as the primary sampling units (PSUs) within the district. From this 2 stage design, we could further sample classrooms within the school (3 stage design) or simply sample students (or whatever our unit of interest is).

A second feature of survey data we often want to account for is differential respondent weighting. This means that each respondent is given a weight to represent how common that particular respondent is within the population. This reflects the differential probability of sampling based on respondent characteristics. As demographers, we are also often interested in making inference for the population, not just the sample, so our results must be generalizable to the population at large. Sample weights are used in the process as well.

When such data are analyzed, we must take into account this nesting structure (sample design) as well as the respondent sample weight in order to make valid estimates of **ANY** statistical parameter. If we do not account for design, the parameter standard errors will be incorrect, and if we do not account for weighting, the parameters themselves will be incorrect and biased.

In general there are typically three things we need to find in our survey data code books: The sample strata identifier, the sample primary sampling unit identifier (often called a cluster identifier) and the respondent survey weight. These will typically have one of these names and should be easily identifiable in the code book.

Statistical software will have special routines for analyzing these types of data and you must be aware that the diversity of statistical routines that generally exists will be lower for analyzing complex survey data, and some forms of analysis *may not be available!*

In the DHS [Recode manual](#), the sampling information for the data is found in variables v021 and v022, which are the primary sampling unit (PSU) and sample strata, respectively. The person weight is found in variable v005, and following DHS protocol, this has six implied decimal places, so we must divide it by 1000000, again, following the DHS manual.

```
dat$wt<-dat$v005/1000000

#create the design: ids == PSU, strata==strata, weights==weights.
options(survey.lonely.psu = "adjust")
des<-svydesign(ids=~v021,
              strata = ~v022,
```

```

        weights=~wt,
        data=dat)

fit.s<-svykm(Surv(death.age, d.event)~rural,
            design=des, se=T)

#use svyby to find the %of infants that die before age 1, by rural/urban status
svyby(~d.event, ~rural, des, svymean)

```

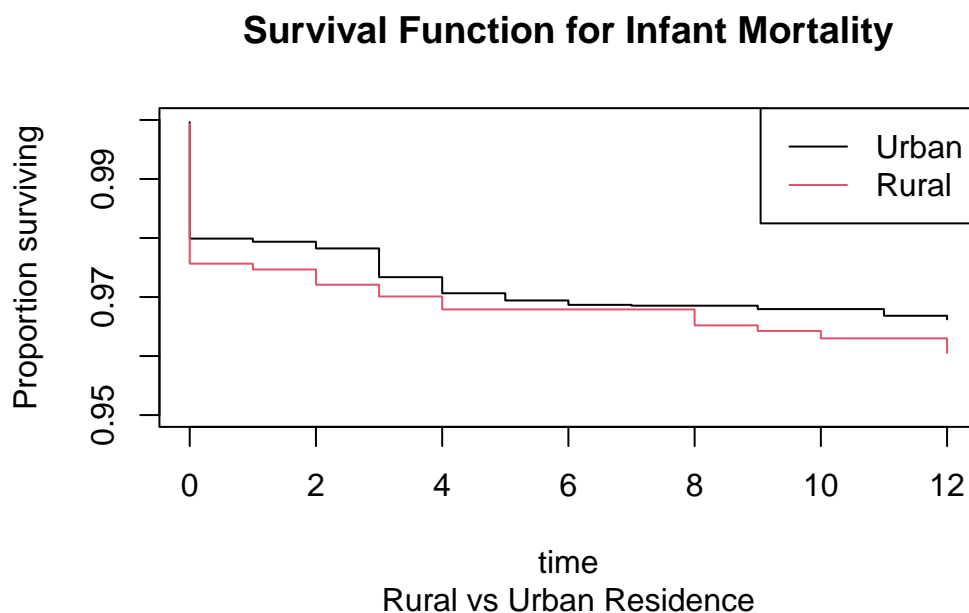
	rural	d.event	se
0rural	0rural	0.03832854	0.005104346
1urban	1urban	0.03305576	0.006363194

The plotting is a bit more of a challenge, as the survey version of the function isn't as nice

```

plot(fit.s[[2]], ylim=c(.95,1), xlim=c(0,12),col=1, ci=F )
lines(fit.s[[1]], col=2)
title(main="Survival Function for Infant Mortality",
      sub="Rural vs Urban Residence")
legend("topright",
      legend = c("Urban","Rural" ),
      col=c(1,2), lty=1)

```



```
#test statistic
svylogrank(Surv(death.age, d.event)~rural, design=des)
```

Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):  
collapsing to unique 'x' values

```
[[1]]
      score
[1,] -4.435842 6.790287 -0.6532629 0.5135868

[[2]]
      Chisq      p
0.4267524 0.5135868

attr(,"class")
[1] "svylogrank"
```

And we see the p-value is larger than assuming random sampling.

## Using a longitudinal data source

In this example, we will examine how to calculate the survival function for a longitudinally collected data set. Here I use data from the [ECLS-K](#).

Exhibit 1-1. Data collection schedule: School years 2010–11 through 2015–16

School year	Grade <sup>1</sup>	Data collections <sup>2</sup>
2010–11	Kindergarten	Fall 2010 Spring 2011
2011–12	First grade	Fall 2011 Spring 2012
2012–13	Second grade	Fall 2012 Spring 2013
2013–14	Third grade	Spring 2014
2014–15	Fourth grade	Spring 2015
2015–16	Fifth grade	Spring 2016

<sup>1</sup> Grade indicates the model grade for children who were in kindergarten in the 2010–11 school year. After the kindergarten rounds of data

Specifically, we will examine the transition into poverty between kindergarten and third grade.

First we load our data

```
eclskk5<-readRDS("C:/Users/ozd504/OneDrive - University of Texas at San Antonio/classes/de
names(eclskk5)<-tolower(names(eclskk5))

#get out only the variables I'm going to use for this example
myvars<-c( "childid","x_chsex_r", "x_raceth_r", "x1kage_r","x4age",
          "x5age", "x6age", "x7age", "x2povty","x4povty_i", "x6povty_i",
          "x8povty_i","x12parled_i", "s2_id")
eclskk5<-eclskk5[,myvars]
```

Recode variables:

```
# time varying variables
eclskk5$age_1<-ifelse(eclskk5$x1kage_r== -9, NA, eclskk5$x1kage_r/12)
eclskk5$age_2<-ifelse(eclskk5$x4age== -9, NA, eclskk5$x4age/12)
#for the later waves, the NCES group the ages into ranges of months,
#so 1= <105 months, 2=105 to 108 months.
#So, I fix the age at the midpoint of the interval they give,
#and make it into years by dividing by 12

eclskk5$age_3<-ifelse(eclskk5$x5age== -9, NA, eclskk5$x5age/12)

eclskk5$pov_1<-ifelse(eclskk5$x2povty==1,1,0)
eclskk5$pov_2<-ifelse(eclskk5$x4povty_i==1,1,0)
eclskk5$pov_3<-ifelse(eclskk5$x6povty_i==1,1,0)

#Time constant variables
#Recode race with white, non Hispanic as reference using dummy vars
eclskk5$race_rec<-Recode (eclskk5$x_raceth_r, recodes="1 = 'nhwhite';2='nhblack';3:4='hisp'
                        as.factor = T)
eclskk5$male<-Recode(eclskk5$x_chsex_r, recodes="1=1; 2=0; -9=NA")
eclskk5$mlths<-Recode(eclskk5$x12parled_i, recodes = "1:2=1; 3:9=0; else = NA")
eclskk5$mgths<-Recode(eclskk5$x12parled_i, recodes = "1:3=0; 4:9=1; else =NA")
```

**NOTE** I need to remove any children who are missing any of the necessary variables, and who are already in poverty in wave 1, because they are not at risk of experiencing **this particular** transition.

Again, this is called forming the *risk set*

```

eclskk5<-eclskk5 %>% filter(is.na(pov_1)==F &
                           is.na(pov_2)==F &
                           is.na(pov_3)==F &
                           is.na(age_1)==F &
                           is.na(age_2)==F &
                           is.na(age_3)==F &
                           pov_1!=1)%>%
mutate(povtran_1 =ifelse(pov_1==0 & pov_2==0, 0,1),
       povtran_2 = ifelse(povtran_1==1, NA,
                           ifelse(pov_2==0 & pov_3==0,0,1)))

```

Now we do the entire data set. To analyze data longitudinally, we need to reshape the data from the current “wide” format (repeated measures in columns) to a “long” format (repeated observations in rows).

The `pivot_long()` function allows us to do this easily. It allows us to specify our repeated measures, time varying covariates as well as time-constant covariates.

```

e.long1 <- eclskk5 %>%
#rename(wt = w4c4p_40, strata= w4c4p_4str, psu = w4c4p_4psu)%>%
select(childid, male, race_rec, mlths, mgths, #time constant
       age_1, age_2, age_3, #t-varying variables
       pov_1, pov_2, pov_3)%>%
pivot_longer(cols = c(-childid, -male, -race_rec, -mlths, -mgths), #time constant variables
             names_to = c(".value", "wave"), #make wave variable and put t-v vars into
             names_sep = "_") #all t-v variables have _ between name and time, like age_

```

Now, I need to form the transition variable, this is my event variable, and in this case it will be 1 if a child enters poverty between the first wave of the data and the third grade wave, and 0 otherwise.

```

e.long1 <- e.long1%>%
group_by(childid)%>%
mutate(nexpov = dplyr::lead(pov,n=1, order_by = childid))%>%
mutate(povtran = ifelse(nexpov == 1 & pov == 0, 1, 0))

```

```

#find which kids failed in the first time period and remove them from the second risk period
failed1<-which(is.na(e.long1$povtran)==T)
e.long1<-e.long1[-failed1,]

```

```
print(e.long1, n = 27)
```

```
# A tibble: 4,316 x 10
```

```
# Groups:   childid [2,072]
```

	childid	male	race_rec	mlths	mgths	wave	age	pov	nexpov	povtran
	<chr>	<dbl>	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	10000014	1	nhwhite	0	0	1	5.65	0	0	0
2	10000014	1	nhwhite	0	0	2	7.16	0	0	0
3	10000020	0	nhasian	0	0	1	5.70	0	0	0
4	10000020	0	nhasian	0	0	2	7.38	0	0	0
5	10000022	0	other	0	1	1	5.72	0	0	0
6	10000022	0	other	0	1	2	7.31	0	0	0
7	10000029	0	nhwhite	1	0	1	5.78	0	0	0
8	10000029	0	nhwhite	1	0	2	7.24	0	0	0
9	10000034	1	nhblack	0	0	1	6.35	0	0	0
10	10000034	1	nhblack	0	0	2	7.78	0	1	1
11	10000034	1	nhblack	0	0	3	8.30	1	NA	0
12	10000040	0	nhasian	0	1	1	5.36	0	0	0
13	10000040	0	nhasian	0	1	2	6.90	0	0	0
14	10000046	1	nhwhite	0	1	1	5.92	0	0	0
15	10000046	1	nhwhite	0	1	2	7.38	0	0	0
16	10000047	1	nhwhite	0	0	1	5.23	0	0	0
17	10000047	1	nhwhite	0	0	2	6.71	0	0	0
18	10000053	1	nhwhite	0	1	1	5.29	0	0	0
19	10000053	1	nhwhite	0	1	2	6.93	0	0	0
20	10000062	1	hispanic	0	1	1	5.82	0	1	1
21	10000062	1	hispanic	0	1	2	7.21	1	1	0
22	10000062	1	hispanic	0	1	3	7.85	1	NA	0
23	10000066	1	hispanic	0	1	1	5.38	0	0	0
24	10000066	1	hispanic	0	1	2	6.89	0	0	0
25	10000075	1	nhwhite	0	1	1	5.53	0	0	0
26	10000075	1	nhwhite	0	1	2	6.98	0	0	0
27	10000092	1	nhwhite	0	1	1	5.32	0	0	0

```
# ... with 4,289 more rows
```

So, this shows us the repeated measures nature of the longitudinal data set.

```
#poverty transition
fit<-survfit(Surv(time = as.numeric(wave), event = povtran) ~ 1,
             data = e.long1)

fit
```

```
Call: survfit(formula = Surv(time = as.numeric(wave), event = povtran) ~
  1, data = e.long1)
```

```
      n events median 0.95LCL 0.95UCL
[1,] 4316    225     NA      NA      NA
```

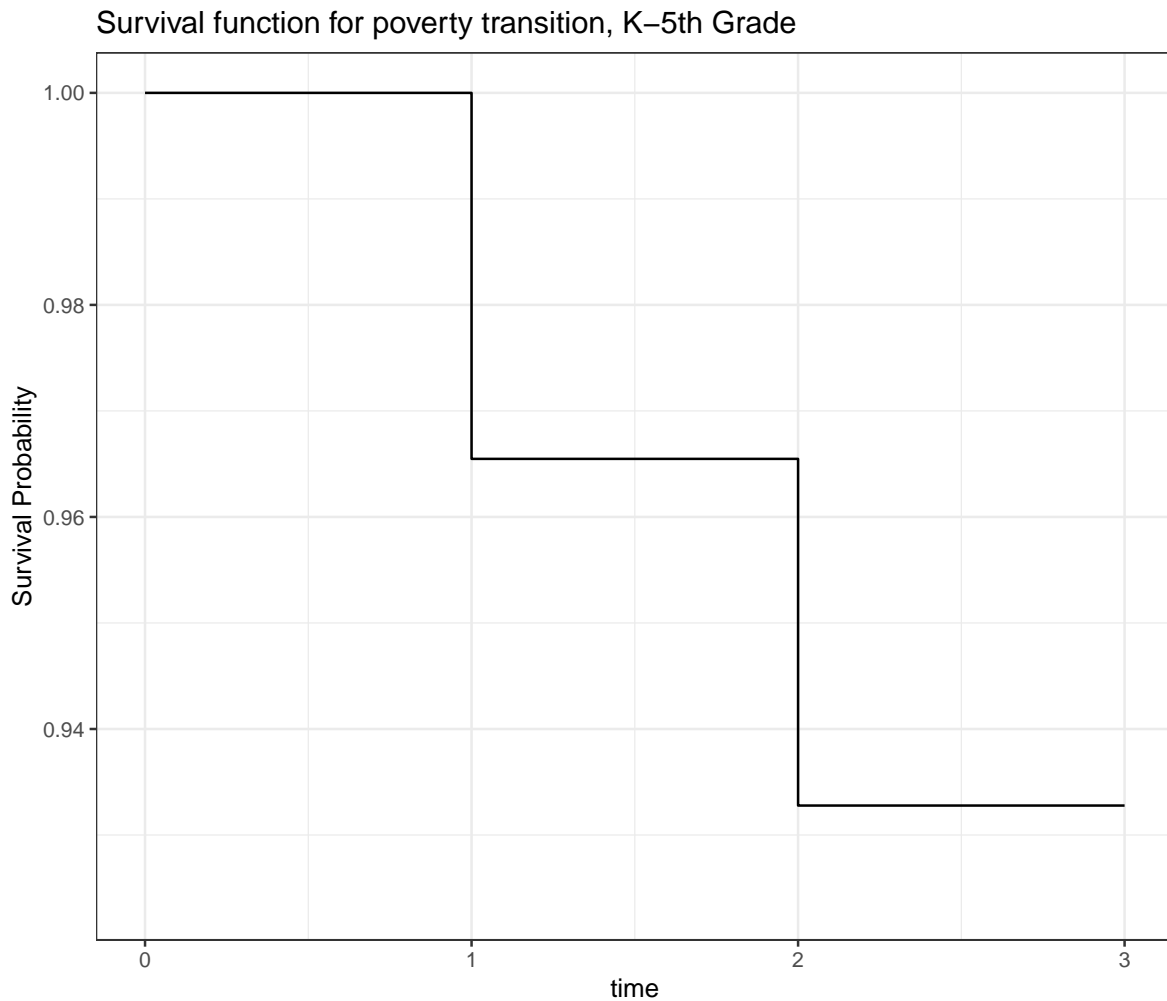
```
summary(fit)
```

```
Call: survfit(formula = Surv(time = as.numeric(wave), event = povtran) ~
  1, data = e.long1)
```

```
time n.risk n.event survival std.err lower 95% CI upper 95% CI
  1   4316    149    0.965 0.00278    0.960    0.971
  2   2244     76    0.933 0.00456    0.924    0.942
```

```
fit %>%
  ggsurvfit(risk.table = T) +
  labs(title="Survival function for poverty transition, K-5th Grade")
```

Warning: Ignoring unknown parameters: risk.table



This displays the number of poverty transitions between each of the waves.

A comparison between groups, based on mom's education level can be done with `survtest()`

```
fit2<- survfit(Surv(time = as.numeric(wave), event = povtran) ~ mlths,  
               data = e.long1)  
  
summary(fit2)
```

```
Call: survfit(formula = Surv(time = as.numeric(wave), event = povtran) ~  
              mlths, data = e.long1)
```

12 observations deleted due to missingness



```

mlths=0
time n.risk n.event survival std.err lower 95% CI upper 95% CI
  1   4019     114    0.972 0.00262      0.967      0.977
  2   2075      56    0.945 0.00429      0.937      0.954

```

```

mlths=1
time n.risk n.event survival std.err lower 95% CI upper 95% CI
  1    285      34    0.881 0.0192      0.844      0.919
  2    162      19    0.777 0.0280      0.724      0.834

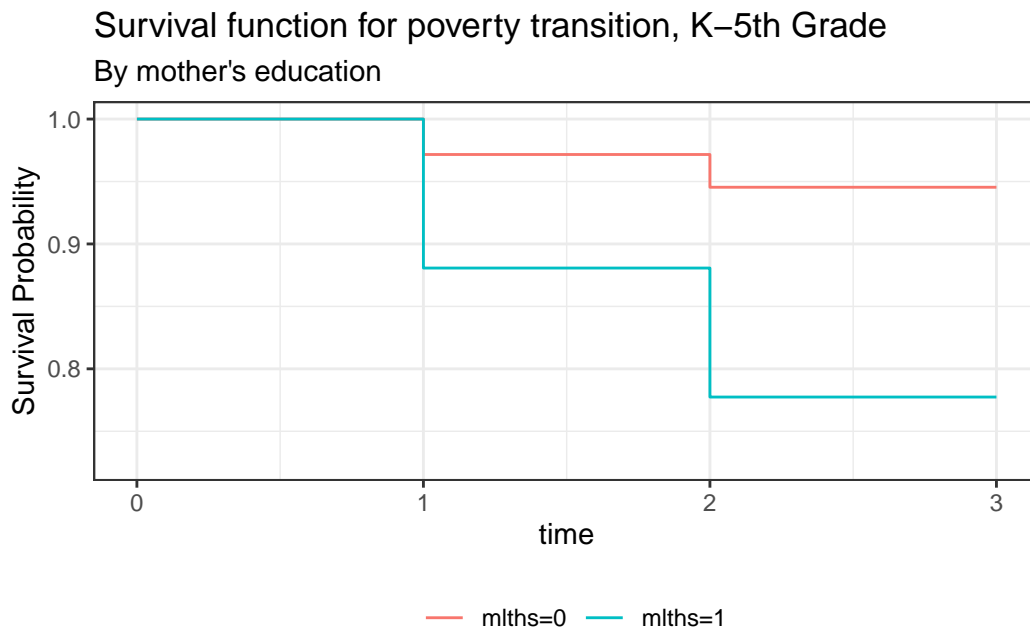
```

```

fit2 %>%
  ggsvrfit(risk.table = T,
            ylim = c(.5, 1)) +
  labs(title="Survival function for poverty transition, K-5th Grade",
        subtitle = "By mother's education" )

```

Warning: Ignoring unknown parameters: risk.table, ylim



```

survdif(Surv(time = as.numeric(wave), event = povtran) ~ mlths,
        data = e.long1)

```

Call:

```
survdif(formula = Surv(time = as.numeric(wave), event = povtran) ~  
  mlths, data = e.long1)
```

n=4304, 12 observations deleted due to missingness.

	N	Observed	Expected	$(O-E)^2/E$	$(O-E)^2/V$
mlths=0	4019	170	207.8	6.87	104
mlths=1	285	53	15.2	93.65	104

Chisq= 104 on 1 degrees of freedom, p= <2e-16