Chapter 13

Miles A. Peña

DSC 530

02/25/2024

Exercise 13-1: In NSFG Cycles 6 and 7, the variable `cmdivorcx` contains the date of divorce for the respondent's first marriage, if applicable, encoded in century-months. Compute the duration of marriages that have ended in divorce, and the duration, so far, of marriages that are ongoing. Estimate the hazard and survival function for the duration of marriage. Use resampling to take into account sampling weights, and plot data from several resamples to visualize sampling error. Consider dividing the respondents into groups by decade of birth, and possibly by age at first marriage.

In [105…
```
from __future__ import print_function
import pandas
import numpy as np
import thinkstats2
import thinkplot
import survival
```

In [72]:
```
def CleanData(resp):

    resp.cmdivorcx.replace([9998, 9999], np.nan, inplace=True)

    resp["notdivorced"] = resp.cmdivorcx.isnull().astype(int)
    resp["duration"] = (resp.cmdivorcx - resp.cmmarrhx) / 12.0
    resp["durationsofar"] = (resp.cmintvw - resp.cmmarrhx) / 12.0

    month0 = pandas.to_datetime("1899-12-15")
    dates = [month0 + pandas.DateOffset(months=cm) for cm in resp.cmbirth]
    resp["decade"] = (pandas.DatetimeIndex(dates).year - 1900) // 10
```

In [73]:
```
resp6 = survival.ReadFemResp2002()
CleanData(resp6)
married6 = resp6[resp6.evrmarry == 1]
```

In [74]:
```
resp7 = survival.ReadFemResp2010()
CleanData(resp7)
married7 = resp7[resp7.evrmarry == 1]
```

In [75]:
```
married6
```

Out[75]:

	caseid	cmbirth	evrmarry	cmmarrhx	cmdivorcx	parity	finalwgt	cmintvw	agemarry	age	year	decade	fives	notdivorced	duration	durationsofar
1	5012	718	True	974.0	1077.0	1	4744.191350	1233	21.333333	42.916667	59	5	11	0	8.583333	21.583333
2	11586	708	True	910.0	938.0	1	4744.191350	1234	16.833333	43.833333	58	5	11	0	2.333333	27.000000
5	845	727	True	937.0	NaN	6	4705.681352	1234	17.500000	42.250000	60	6	12	1	NaN	24.750000
8	8656	780	True	1003.0	NaN	3	6520.021223	1237	18.583333	38.083333	64	6	12	1	NaN	19.500000
10	5917	714	True	953.0	NaN	2	3488.586646	1233	19.916667	43.250000	59	5	11	1	NaN	23.333333
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
7634	1282	798	True	1057.0	1119.0	2	4055.209574	1228	21.583333	35.833333	66	6	13	0	5.166667	14.250000
7635	2954	862	True	1069.0	NaN	1	4087.693768	1228	17.250000	30.500000	71	7	14	1	NaN	13.250000
7636	4964	727	True	953.0	NaN	6	3703.220316	1227	18.833333	41.666667	60	6	12	1	NaN	22.833333
7637	143	808	True	1060.0	1151.0	0	4496.050707	1230	21.000000	35.166667	67	6	13	0	7.583333	14.166667
7638	11018	811	True	1032.0	1053.0	0	6565.818007	1228	18.416667	34.750000	67	6	13	0	1.750000	16.333333

4126 rows × 16 columns

In [76]:
```
married7
```

Out[76]:

	caseid	cmbirth	evrmarry	cmmarrhx	cmdivorcx	parity	wgtq1q16	cmintvw	finalwgt	agemarry	age	year	decade	fives	notdivorced	duration	durationsofar
1	40081	925	True	1314.0	NaN	0	11716.317848	1323	11716.317848	32.416667	33.166667	77	7	15	1	NaN	0.750000
18	33303	773	True	1076.0	NaN	3	6603.626644	1289	6603.626644	25.250000	43.000000	64	6	12	1	NaN	17.750000
19	38594	796	True	1089.0	NaN	4	14915.930053	1325	14915.930053	24.416667	44.083333	66	6	13	1	NaN	19.666667
22	28488	810	True	1126.0	NaN	2	6114.806526	1289	6114.806526	26.333333	39.916667	67	6	13	1	NaN	13.583333
23	37022	866	True	1257.0	NaN	3	7831.375643	1326	7831.375643	32.583333	38.333333	72	7	14	1	NaN	5.750000
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
12270	26745	812	True	1155.0	NaN	2	5220.893454	1282	5220.893454	28.583333	39.166667	67	6	13	1	NaN	10.583333
12271	26914	985	True	1221.0	NaN	2	5152.498030	1282	5152.498030	19.666667	24.750000	82	8	16	1	NaN	5.083333
12273	30517	810	True	1107.0	NaN	3	2830.681441	1282	2830.681441	24.750000	39.333333	67	6	13	1	NaN	14.583333
12275	26372	771	True	1025.0	NaN	4	19328.779624	1286	19328.779624	21.166667	42.916667	64	6	12	1	NaN	21.750000
12277	29718	977	True	1210.0	NaN	3	4876.196382	1282	4876.196382	19.416667	25.416667	81	8	16	1	NaN	6.000000

5534 rows × 17 columns

In [106…
```
def ResampleDivorceCurve(resps):

    for _ in range(41):
        samples = [thinkstats2.ResampleRowsWeighted(resp)
                   for resp in resps]
        sample = pandas.concat(samples, ignore_index = True)
        PlotDivorceCurveByDecade(sample, color='#225EA8', alpha=0.1)

    thinkplot.Show(xlabel='years',
                   axis=[0, 28, 0, 1])
```

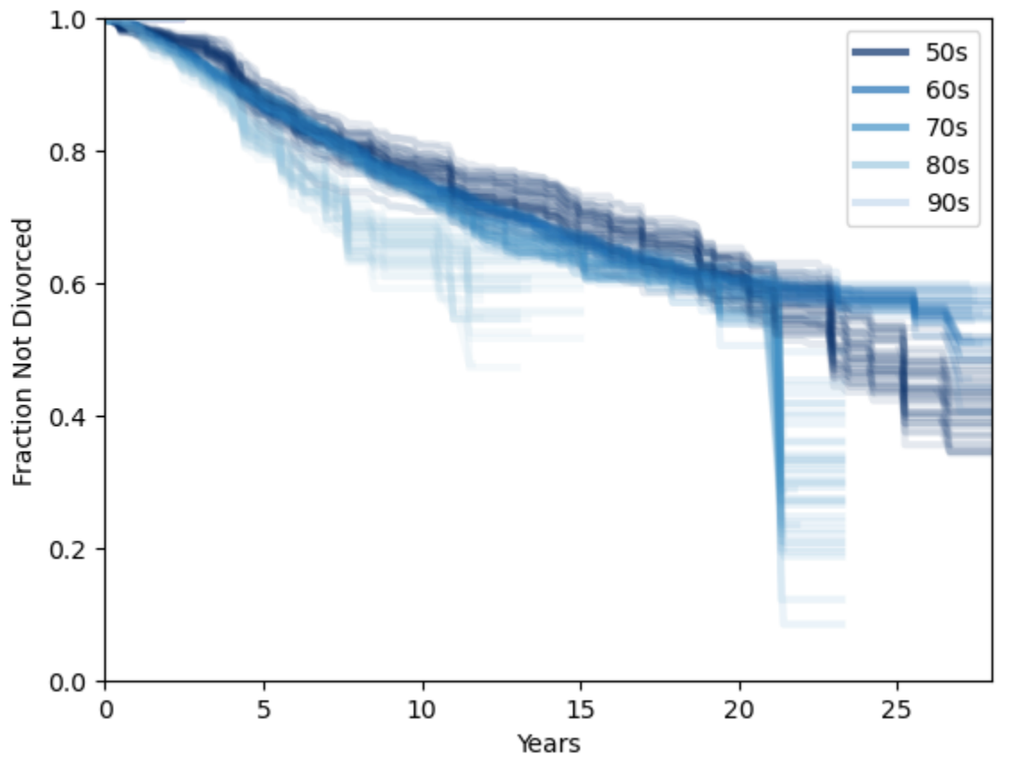
In [46]:
```
def ResampleDivorceCurveByDecade(resps):

    for i in range(41):
        samples = [thinkstats2.ResampleRowsWeighted(resp) for resp in resps]
        sample = pandas.concat(samples, ignore_index = True)
        groups = sample.groupby("decade")
        if i == 0:
            survival.AddLabelsByDecade(groups, alpha=0.7)

        EstimateSurvivalByDecade(groups, alpha=0.1)

    thinkplot.Config(xlabel = "Years", ylabel = "Fraction Not Divorced", axis = [0, 28, 0, 1])
```

In [53]:
```
ResampleDivorceCurveByDecade([married6, married7])
```



In [47]:
```
def EstimateSurvivalByDecade(groups, ** options):

    thinkplot.PrePlot(len(groups))
    for name, group in groups:
        _, sf = EstimateSurvival(group)
        thinkplot.Plot(sf, ** options)
```

In [108…
```
def EstimateSurvival(resp):

    complete = resp[resp.notdivorced == 0].duration.dropna()
    ongoing = resp[resp.notdivorced == 1].durationsofar.dropna()

    hf = survival.EstimateHazardFunction(complete, ongoing)
    sf = hf.MakeSurvival()

    return hf, sf
```