Reppeto530Week11

February 25, 2024

0.1 Chapter 13

Brian Reppeto 530 Prof. Jim Week 11 HW 13-1

[1]:
```
# import libr and .py files

import pandas as pd
import thinkplot
import numpy as np
import nsfg
import survival
import thinkstats2
```

[2]:
```
# using the survival.py file read the repond files as resp

resp6 = survival.ReadFemResp2002()
resp7 = survival.ReadFemResp2010()
```

[3]:
```
# create a funciton to clean and preprocess the data for analysis

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

[4]:
```
#  use the cleandata function on the dataset resp6 to include only those␣
↪respondents who have been married

CleanData(resp6)
married6=resp6[resp6.evrmarry == 1]
```

[5]:
```
# use the cleandata function on the dataset resp7 to include only those␣
↪respondents who have been married

CleanData(resp7)
married7=resp7[resp7.evrmarry == 1]
```

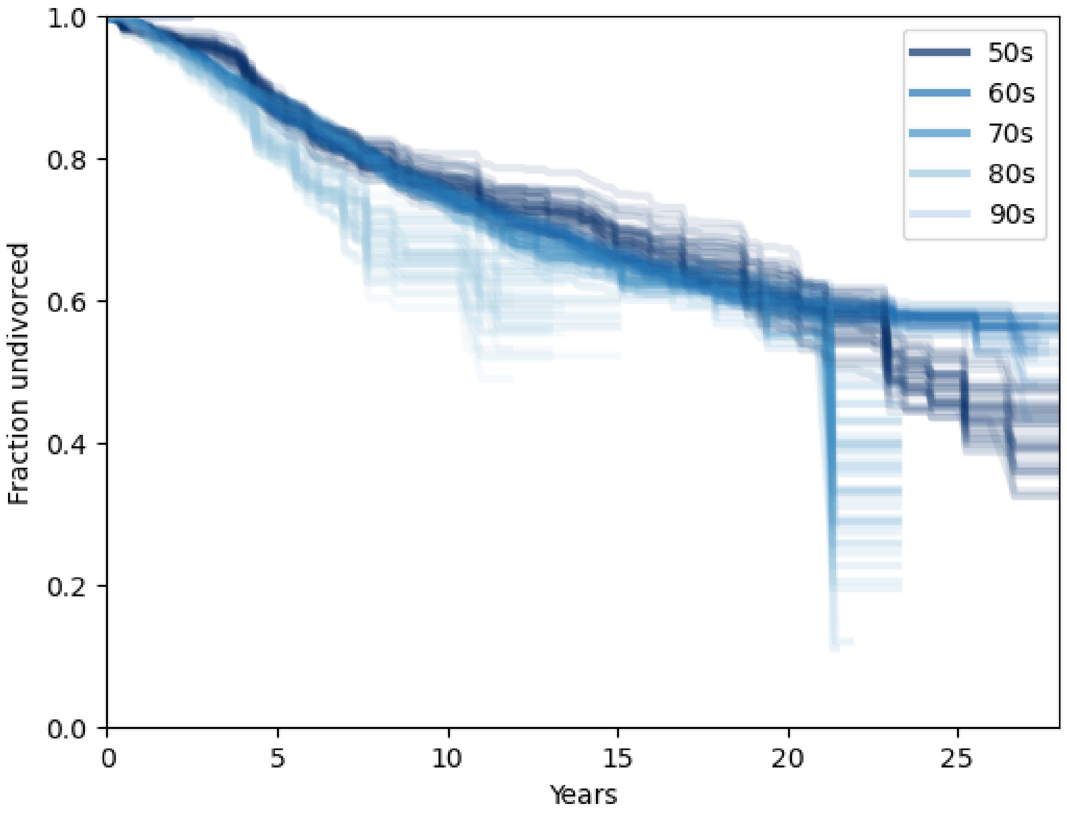
[6]:
```
# create a function to plot divorce curves by decade for a set of respondent␣
↪data

def ResampleDivorceCurve(resps):

    for _ in range(11):
        samples = [thinkstats2.ResampleRowsWeighted(resp) for resp in resps]
        sample = pd.concat(samples, ignore_index=True)
        PlotDivorceCurveByDecade(sample, color="#225EA8", alpha=0.1)

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

[7]:
```
# create a function  to estimate and plot survival curve by decade

def ResampleDivorceCurveByDecade(resps):

    for i in range(41):
        samples = [thinkstats2.ResampleRowsWeighted(resp) for resp in resps]
        sample = pd.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 undivorced", axis=[0, 28,␣
↪0, 1])
```

[8]:
```
# create a function to help estimate and plot survival curves for different␣
↪groups

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

[9]:
```
# create a functin top estimate the survival function about marital status

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
```

[10]:
```
# call the ResampleDivorceCurveByDecade function for the married6 and married7␣
  ↪datasets

ResampleDivorceCurveByDecade([married6, married7])
```



[ ]:
```
```

[ ]:
```
```