C2_W3_lecture

January 5, 2021

1 AI4M Course 2 Week 3 lecture notebook

1.1 Outline

Count patients

[1]:
```
import numpy as np
import pandas as pd
```

We'll work with data where: - Time: days after a disease is diagnosed and the patient either dies or left the hospital's supervision. - Event: - 1 if the patient died - 0 if the patient was not observed to die beyond the given 'Time' (their data is censored)

Notice that these are the same numbers that you see in the lecture video about estimating survival.

[2]:
```
df = pd.DataFrame({'Time': [10,8,60,20,12,30,15],
                   'Event': [1,0,1,1,0,1,0]
                  })
df
```

[2]:

	Time	Event
0	10	1
1	8	0
2	60	1
3	20	1
4	12	0
5	30	1
6	15	0

1.1.1 Count patients

1.1.2 Count number of censored patients

[3]:
```
df['Event'] == 0
```

[3]:
```
0    False
1     True
2    False
3    False
4     True
5    False
6     True
Name: Event, dtype: bool
```

Patient 1, 4 and 6 were censored.

- Count how many patient records were censored

When we sum a series of booleans, `True` is treated as 1 and `False` is treated as 0.

[4]:
```
sum(df['Event'] == 0)
```

[4]:
```
3
```

1.1.3 Count number of patients who definitely survived past time t

This assumes that any patient who was censored died at the time of being censored (**died immediately**).

If a patient survived past time `t`: - Their `Time` of event should be greater than `t`.
- Notice that they can have an `Event` of either 1 or 0. What matters is their `Time` value.

[5]:
```
t = 25
df['Time'] > t
```

[5]:
```
0    False
1    False
2     True
3    False
4    False
5     True
6    False
Name: Time, dtype: bool
```

[6]:
```
sum(df['Time'] > t)
```

[6]:
```
2
```

1.1.4 Count the number of patients who may have survived past t

This assumes that censored patients **never die**. - The patient is censored at any time and we assume that they live forever. - The patient died (`Event` is 1) but after time `t`

[7]:
```
t = 25
(df['Time'] > t) | (df['Event'] == 0)
```

[7]:
```
0    False
1     True
2     True
3    False
4     True
5     True
6     True
dtype: bool
```

[8]:
```
sum( (df['Time'] > t) | (df['Event'] == 0) )
```

[8]:
```
5
```

1.1.5 Count number of patients who were not censored before time t

If patient was not censored before time `t`: - They either had an event (death) before `t`, at `t`, or after `t` (any time) - Or, their `Time` occurs after time `t` (they may have either died or been censored at a later time after `t`)

[9]:
```
t = 25
(df['Event'] == 1) | (df['Time'] > t)
```

[9]:
```
0     True
1    False
2     True
3     True
4    False
5     True
6    False
dtype: bool
```

[10]:
```
sum( (df['Event'] == 1) | (df['Time'] > t) )
```

[10]:
```
4
```

Kaplan-Meier

The Kaplan Meier estimate of survival probability is:

$$S(t) = \prod_{t_i \leq t} (1 - \frac{d_i}{n_i})$$

- t_i are the events observed in the dataset
- d_i is the number of deaths at time t_i
- n_i is the number of people who we know have survived up to time t_i.

[11]:
```
import numpy as np
import pandas as pd
```

[12]:
```
df = pd.DataFrame({'Time': [3,3,2,2],
                   'Event': [0,1,0,1]
                  })
df
```

[12]:

	Time	Event
0	3	0
1	3	1
2	2	0
3	2	1

1.1.6 Find those who survived up to time t_i

If they survived up to time t_i, - Their `Time` is either greater than t_i - Or, their `Time` can be equal to t_i

[13]:
```
t_i = 2
df['Time'] >= t_i
```

[13]:
```
0    True
1    True
2    True
3    True
Name: Time, dtype: bool
```

You can use this to help you calculate n_i

1.1.7 Find those who died at time t_i

- If they died at t_i:
- Their `Event` value is 1.

- Also, their `Time` should be equal to t_i

[14]:
```
t_i = 2
(df['Event'] == 1) & (df['Time'] == t_i)
```

[14]:
```
0    False
1    False
2    False
3     True
dtype: bool
```

You can use this to help you calculate d_i

You'll implement Kaplan Meier in this week's assignment!