

```
In [1]: 1 import pandas as pd
        2 import numpy as np
        3 import matplotlib.pyplot as plt
        4 from lifelines import KaplanMeierFitter
```

```
In [2]: 1 data = pd.read_csv("lung.csv")
        2 data.head()
```

```
Out[2]:
```

	Unnamed: 0	inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss
0	1	3.0	306	2	74	1	1.0	90.0	100.0	1175.0	NaN
1	2	3.0	455	2	68	1	0.0	90.0	90.0	1225.0	15.0
2	3	3.0	1010	1	56	1	0.0	90.0	90.0	NaN	15.0
3	4	5.0	210	2	57	1	1.0	90.0	60.0	1150.0	11.0
4	5	1.0	883	2	60	1	0.0	100.0	90.0	NaN	0.0

```
In [3]: 1 data.columns
```

```
Out[3]: Index(['Unnamed: 0', 'inst', 'time', 'status', 'age', 'sex', 'ph.ecog',
              'ph.karno', 'pat.karno', 'meal.cal', 'wt.loss'],
              dtype='object')
```

```
In [ ]:
```

```
1
```

In [4]: 1 data.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 228 entries, 0 to 227
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   Unnamed: 0   228 non-null   int64   
1   inst         227 non-null   float64  
2   time         228 non-null   int64   
3   status       228 non-null   int64   
4   age          228 non-null   int64   
5   sex          228 non-null   int64   
6   ph.ecog      227 non-null   float64  
7   ph.karno     227 non-null   float64  
8   pat.karno    225 non-null   float64  
9   meal.cal     181 non-null   float64  
10  wt.loss      214 non-null   float64  
dtypes: float64(6), int64(5)
memory usage: 19.7 KB

```

In [5]: 1 data.describe()

Out[5]:

	Unnamed: 0	inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss
count	228.00000	227.000000	228.000000	228.000000	228.000000	228.000000	227.000000	227.000000	225.000000	181.000000	214.000000
mean	114.50000	11.088106	305.232456	1.723684	62.447368	1.394737	0.951542	81.938326	79.955556	928.779006	9.831000
std	65.96211	8.303491	210.645543	0.448159	9.073457	0.489870	0.717872	12.327955	14.623177	402.174707	13.131000
min	1.00000	1.000000	5.000000	1.000000	39.000000	1.000000	0.000000	50.000000	30.000000	96.000000	-24.000000
25%	57.75000	3.000000	166.750000	1.000000	56.000000	1.000000	0.000000	75.000000	70.000000	635.000000	0.000000
50%	114.50000	11.000000	255.500000	2.000000	63.000000	1.000000	1.000000	80.000000	80.000000	975.000000	7.000000
75%	171.25000	16.000000	396.500000	2.000000	69.000000	2.000000	1.000000	90.000000	90.000000	1150.000000	15.750000
max	228.00000	33.000000	1022.000000	2.000000	82.000000	2.000000	3.000000	100.000000	100.000000	2600.000000	68.000000

```
In [7]: 1 kmf = KaplanMeierFitter()
```

```
In [8]: 1 data.loc[data.status == 1, 'dead'] = 0
        2 data.loc[data.status == 2, 'dead'] = 1
```

```
In [9]: 1 data.head()
```

```
Out[9]:
```

	Unnamed: 0	inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss	dead
0	1	3.0	306	2	74	1	1.0	90.0	100.0	1175.0	NaN	1.0
1	2	3.0	455	2	68	1	0.0	90.0	90.0	1225.0	15.0	1.0
2	3	3.0	1010	1	56	1	0.0	90.0	90.0	NaN	15.0	0.0
3	4	5.0	210	2	57	1	1.0	90.0	60.0	1150.0	11.0	1.0
4	5	1.0	883	2	60	1	0.0	100.0	90.0	NaN	0.0	1.0

```
In [10]: 1 ##Fitting our data
        2 kmf.fit(durations = data["time"], event_observed = data["dead"])
```

```
Out[10]: <lifelines.KaplanMeierFitter:"KM_estimate", fitted with 228 total observations, 63 right-censored observations
>
```

In [11]: 1 kmf.event_table

Out[11]:

	removed	observed	censored	entrance	at_risk
event_at					
0.0	0	0	0	228	228
5.0	1	1	0	0	228
11.0	3	3	0	0	227
12.0	1	1	0	0	224
13.0	2	2	0	0	223
...
840.0	1	0	1	0	5
883.0	1	1	0	0	4
965.0	1	0	1	0	3
1010.0	1	0	1	0	2
1022.0	1	0	1	0	1

187 rows × 5 columns

In [16]: 1 *##Prediction*
2 kmf.predict(11)

Out[16]: 0.9824561403508766

In [17]: 1 kmf.predict([0,5,11,12])

Out[17]: 0 1.000000
5 0.995614
11 0.982456
12 0.978070
Name: KM_estimate, dtype: float64

```
In [18]: 1 kmf.survival_function_
```

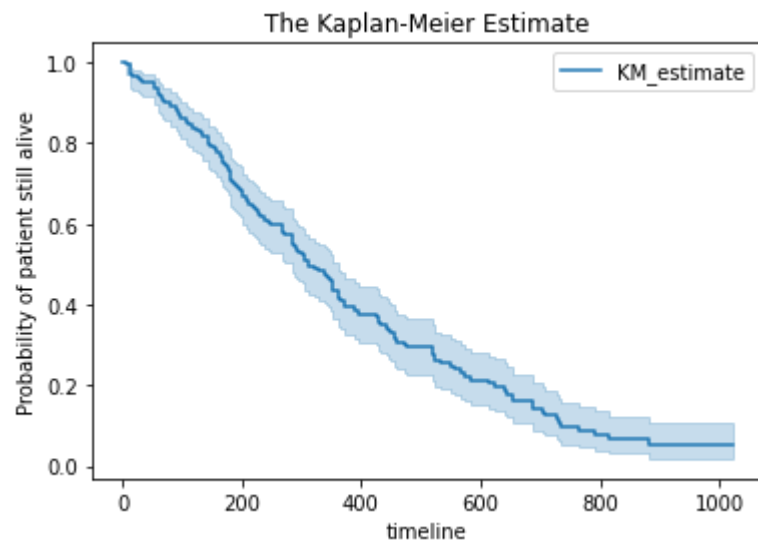
```
Out[18]:
```

KM_estimate	
timeline	
0.0	1.000000
5.0	0.995614
11.0	0.982456
12.0	0.978070
13.0	0.969298
...	...
840.0	0.067127
883.0	0.050346
965.0	0.050346
1010.0	0.050346
1022.0	0.050346

timeline	
0.0	1.000000
5.0	0.995614
11.0	0.982456
12.0	0.978070
13.0	0.969298
...	...
840.0	0.067127
883.0	0.050346
965.0	0.050346
1010.0	0.050346
1022.0	0.050346

187 rows × 1 columns

```
In [19]: 1 kmf.plot()  
2 plt.title("The Kaplan-Meier Estimate")  
3 plt.ylabel("Probability of patient still alive")  
4 plt.show()
```



```
In [ ]:
```

```
1
```