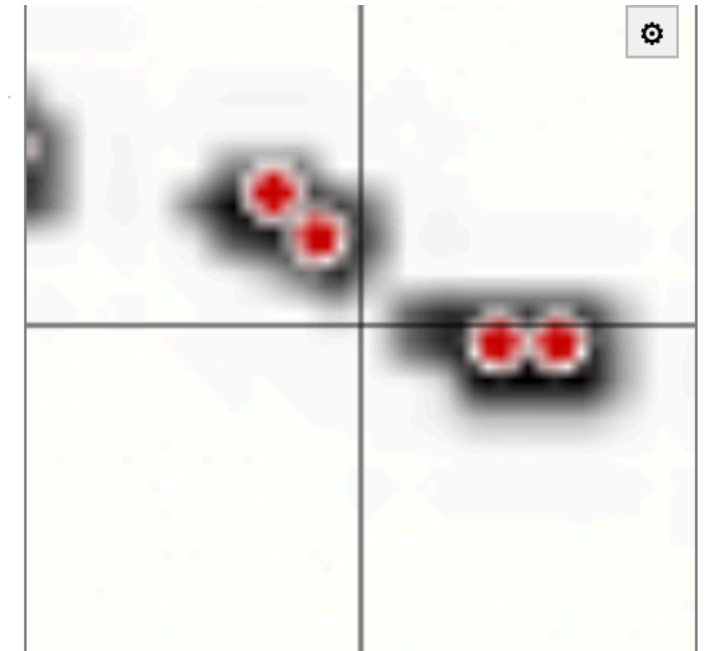
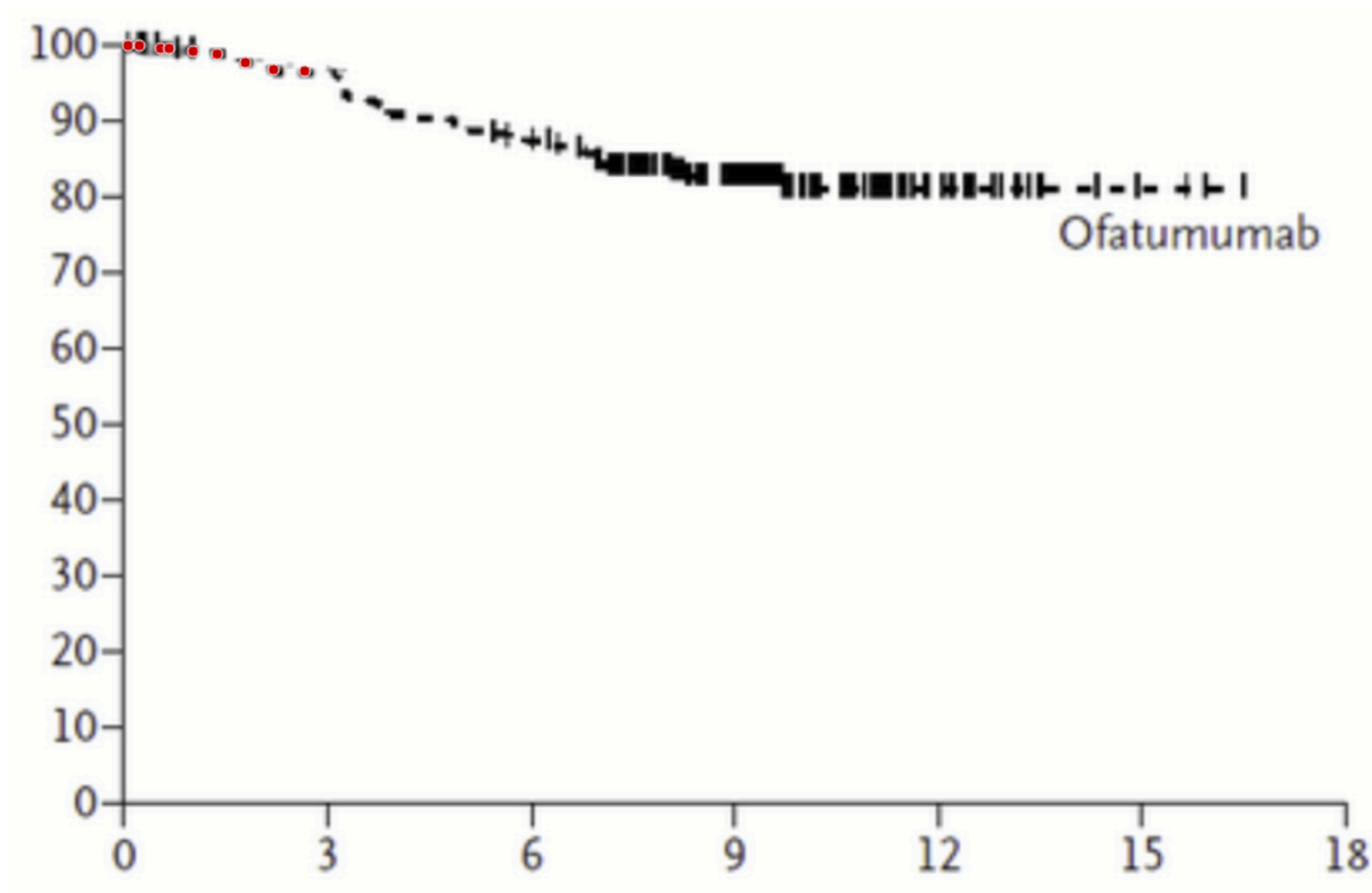


AŞAMALAR

1. Grafikteki Verileri Csv Formatına Getirme
2. Veriyi Yükleme
3. Bireysel Hasta Düzeyinde Veri Oluşturma
4. Farklı Olasılık Dağılımları Uydurma
5. En Uygun Dağılımı Bulma
6. Zamana Bağımlı Ölüm İlerlemesi Olasılıklarının Oluşturulması
7. Verileri Kontrol Etme/
8. Sonuçların Kaydedilmesi

Grafikten verileri alma

- WebPlotDigitizer kullanarak grafikten verileri okuduk.



Overall_Survival verilerimiz

- Hastalık durumunda hayatta kalma oranını ifade eder.

Overall_Survival.csv

1 to 62 of 62 entries

Months	Overall_survival
0	99.771
0	99.771
1	99.771
1	100.0
1	99.541
1	98.853
2	97.706
2	96.56
3	96.33
3	96.101
3	93.349
4	92.431
4	90.826
4	90.367
5	89.908
5	88.532
5	88.532
6	88.073
6	87.385
6	86.697
6	87.156
7	86.239
7	85.78
7	84.404


Progression_Free_Survival Verilerimiz

- Hastalığın zaman içinde nasıl ilerleyeceğini ve tedaviye nasıl yanıt vereceğini anlamak için kullanılır.

Progression_Free_Survival.csv ✕	
Months	Progression-free_survival
1	99.773
1	98.866
1	97.506
1	95.918
2	95.465
2	93.651
2	92.29
3	91.156
3	90.249
3	89.569
3	87.302
3	85.941
4	84.127
4	82.766
4	81.859
4	80.952
5	80.272
5	79.365
5	78.458
5	77.551
5	75.964
5	74.15
5	72.789
5	71.429

Kütüphaneleri yükleme / verileri indirme / sütun isimlerini düzeltme

```
✓  
0  
sn. [73] import pandas as pd  
      import numpy as np  
      from scipy.stats import expon, weibull_min, gamma, lognorm
```

```
✓  
0  
sn.  pfs_data = pd.read_csv("Progression_Free_Survival.csv")  
    os_data = pd.read_csv("Overall_Survival.csv")
```

```
✓  
0  
sn. [35] pfs_data.columns = ["Months", "Progression_Free_Survival"]  
      os_data.columns = ["Months", "Overall_Survival"]
```

Bireysel hasta düzeyinde veri oluşturma

- Bu döngü, her bir hastayı temsil etmek için "survival_rate" değeri kadar kez çalışıyor. Bu döngü her çalıştığında, hastanın yaşadığı ay (months) individual_data listesine ekleniyor.
- Bu, her bir hastanın sürekli olarak yaşadığı ayları temsil etmek için yapılan bir işlemdir.

✓
0
sn.



```
# Bireysel hasta düzeyinde veri oluşturma fonksiyonu
def generate_individual_data(data, survival_type):
    individual_data = []
    for index, row in data.iterrows():
        months = row['Months']
        survival_rate = row[survival_type] / 100 # Yüzdeyi olasılığa dönüştür
        for i in range(int(survival_rate*100)):
            individual_data.append(months)
    return pd.DataFrame({'Months': individual_data})

individual_pfs_data = generate_individual_data(pfs_data, 'Progression_Free_Survival')
individual_os_data = generate_individual_data(os_data, 'Overall_Survival')
```

En Uygun Dağılımı Bulma

- `fit_distribution` adlı bir fonksiyon tanımlanır. Bu fonksiyon, bir veri kümesi alır ve bu veri kümesi için belirli bir dizi olasılık dağılımını (exponential, Weibull, gamma, lognormal gibi) uygun hale getirir. Fonksiyon, en iyi uyum sağlayan parametreleri döndürür.
- Sonuç olarak, bu fonksiyon, veri kümesi için farklı olasılık dağılımlarını uygun hale getirir ve her bir dağılım için en iyi uyum sağlayan parametreleri döndürür.

✓
0
sn.

```
[82] def fit_distribution(data):  
    distributions = {'expon': expon, 'weibull_min': weibull_min, 'gamma': gamma, 'lognorm': lognorm}  
    best_params = {}  
    for name, dist in distributions.items():  
        params = dist.fit(data)  
        best_params[name] = params  
    return best_params  
  
pfs_params = fit_distribution(individual_pfs_data)  
os_params = fit_distribution(individual_os_data)
```


Zamana Bağımlı Ölüm İlerlemesi

- Kod, verilen veri ve uyumlanan dağılım parametreleri kullanılarak her bir zaman noktası için ölüm ilerleme olasılıklarını hesaplar ve bir veri çerçevesi olarak döndürür. Bu veri çerçeveleri daha sonra analiz için kullanılabilir veya başka bir amaç için kaydedilebilir.

```
✓ [83] # İstenilen 250 aylık period için ölüm ilerlemesi olasılıkları oluşturma
0
sn.

def generate_death_progression_probabilities(data, params, period=251):
    time_points = list(range(0, period + 1))
    probabilities = []
    for t in time_points:
        prob = 0
        for dist, dist_params in params.items():
            if dist == 'expon':
                prob += expon.pdf(t, *dist_params)
            elif dist == 'weibull_min':
                prob += weibull_min.pdf(t, *dist_params)
            elif dist == 'gamma':
                prob += gamma.pdf(t, *dist_params)
            elif dist == 'lognorm':
                prob += lognorm.pdf(t, *dist_params)
        probabilities.append(prob)
    return pd.DataFrame({'Time': time_points, 'Probability': probabilities})

pfs_death_probabilities = generate_death_progression_probabilities(individual_pfs_data['Months'], pfs_params)
os_death_probabilities = generate_death_progression_probabilities(individual_os_data['Months'], os_params)
```

VERİYİ DÜZENLEME

```
✓  
0  
sn. [6] # Time da hata aldık ID sütunu eklendi  
      pfs_death_probabilities['ID'] = pfs_death_probabilities.index  
      os_death_probabilities['ID'] = os_death_probabilities.index  
  
      # CSV ye kaydetme  
      pfs_death_probabilities[['ID', 'Probability']].to_csv('sample_submission.csv', index=False)
```

```
✓  
0  
sn. [4] data = pd.read_csv("sample_submission.csv")
```

```
✓  
0  
sn. [5] # ID sütununda değeri 0 olan satırları düşür  
      data = data[data['ID'] != 0]  
  
      # Veri çerçevesinin 'Probability' sütunundaki değerlerin ortalamasını hesapla  
      average_probability = data['Probability'].mean()  
  
      print("Olasılık değerlerinin ortalaması:", average_probability)
```

Olasılık değerlerinin ortalaması: 0.011900126211595684

VERİYİ DÜZENLEME

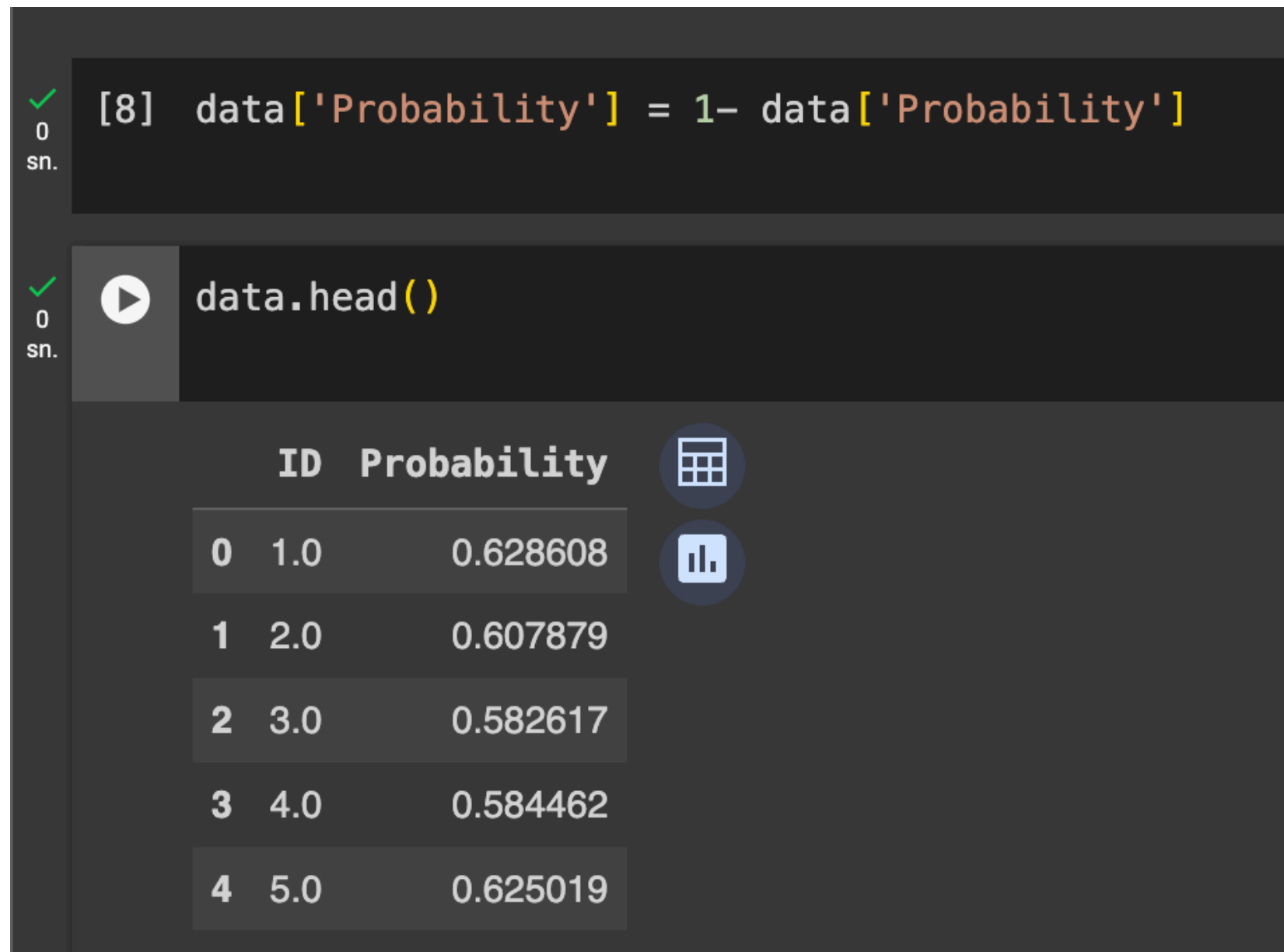
✓
0
sn.



```
new_row = {'ID': 0, 'Probability': average_probability}  
print(average_probability)  
# Yeni satırı ekleyin  
data = data.append(new_row, ignore_index=True)  
data
```

İSTENİLEN DEĞERİ HESAPLAMA

- Bir hastanın belirli bir zaman aralığında ilerlemesiz sağkalımı hesaplayarak elde edilen olay gerçekleşme olasılığından, o zaman aralığında meydana gelen ölüm olasılığının çıkarılması gerekir. Böylece, geriye kalan olasılık, ilerlemenin gerçekleşme olasılığını temsil eder.



The image shows a Jupyter Notebook interface with two code cells. The first cell contains the code `[8] data['Probability'] = 1- data['Probability']` and is marked as successful. The second cell contains the code `data.head()` and is also marked as successful. Below the code cells, the output of `data.head()` is displayed as a table with three columns: **ID**, **Probability**, and an unlabeled column. The table contains five rows of data. To the right of the table, there are two circular icons: a grid icon and a bar chart icon.

	ID	Probability	
0	1.0	0.628608	
1	2.0	0.607879	
2	3.0	0.582617	
3	4.0	0.584462	
4	5.0	0.625019	

VERİYİ DÜZENLEME

✓
0
sn.



```
# Son satırı al
last_row = data.iloc[-1]

# Son satırı veri setinden çıkar
data = data.iloc[:-1]

# Veri setine ilk satırı ekleyin
data = pd.concat([last_row.to_frame().T, data], ignore_index=True)

# Veri setini yeniden indeksleme
data.reset_index(drop=True, inplace=True)

# Veri setini kontrol etmek için yazdırma
print(data)
```

✓
0
sn.



```
# Güncellenmiş veriyi sample_submission.csv dosyasına yaz
data.to_csv("sample_submission.csv", index=False)

print("Veriler başarıyla güncellendi ve sample_submission.csv dosyasına yazıldı.")
```

Veriler başarıyla güncellendi ve sample_submission.csv dosyasına yazıldı.

SONUÇ ÇIKTISI

sample_submission.csv ✕	
ID	Probability
0.0	0.9880998737884044
1.0	0.6286077141676789
2.0	0.6078785613066924
3.0	0.5826166622332019
4.0	0.5844621077554217
5.0	0.6250187435130918
6.0	0.6969741457965681
7.0	0.7807470461532584
8.0	0.8564620800199146
9.0	0.9128745747312466
10.0	0.9487794680123559
11.0	0.9690020880578137
12.0	0.9796182991102363
13.0	0.9852482803584095
14.0	0.9885473481195787
15.0	0.9907639089240503
16.0	0.9924120549690729
17.0	0.9936994683324848
18.0	0.9947239212350101
19.0	0.9955446509151984
20.0	0.9962044555821673
21.0	0.9967364498717685
22.0	0.9971667419666917

sample_submission.csv ✕	
227.0	0.9998730025212982
228.0	0.9998735945806558
229.0	0.9998741813271018
230.0	0.9998747628311079
231.0	0.9998753391619107
232.0	0.999875910387539
233.0	0.9998764765748396
234.0	0.9998770377895029
235.0	0.9998775940960877
236.0	0.9998781455580456
237.0	0.999878692237744
238.0	0.9998792341964899
239.0	0.9998797714945517
240.0	0.999880304191181
241.0	0.9998808323446345
242.0	0.9998813560121941
243.0	0.9998818752501873
244.0	0.9998823901140073
245.0	0.9998829006581317
246.0	0.9998834069361416
247.0	0.9998839090007401
248.0	0.9998844069037696
249.0	0.9998849006962297
250.0	0.9998853904282942

Teşekkürler

HAZIRLAYANLAR:

ZEYNEP KICIKOĞLU

YÜSRA KAPLAN