

Nokta tahminlerini, popülasyon ortalamaları, varyansları ve diğer istatistikleri tahmin etmek için kullanırız. Bu tahminleri elde etmek için, popülasyonumuz için ölçmek istediğimiz fonksiyonu basitçe bir veri örneğine uyguluyoruz.

Örneğin, 9.000 çalışanı olan bir şirket olduğunu ve çalışanların bir gün içinde verdikleri molaların ortalama süresini belirlemek istediğimizi varsayalım. Muhtemelen her bir kişiye bunu soramayacağımız için, 9.000 kişiden bir örneklem alır ve bu örneklemin ortalamasını hesaplarız. Bu örneklem ortalaması, nokta tahminimiz olacaktır.

Rastgele olarak 9.000 kişinin bu soruya verdiği yanıtları üretmek için Poisson dağılımı adı verilen olasılık dağılımını kullanacağız. Yani, "Gün içinde genellikle kaç dakika mola veriyorsunuz?" sorusuna verilen cevapları Poisson dağılımı ile oluşturacağız.

Üretilirliği sağlamak için rastgele bir tohum (seed) belirledim (bu, her seferinde aynı rastgele sayıları elde etmemizi sağlar).

100 çalışanlık bir örneklem alacağız (Python'un `random.sample` metodunu kullanarak) ve bir ortalama nokta tahmini (örneklem ortalaması olarak adlandırılır) bulacağız

Önemli Not

Bu örneklemin, toplam popülasyonun yalnızca %1'inden biraz fazla olduğunu unutmayın. Örneklem ortalamamızı (100 çalışanın ortalaması) popülasyon ortalamasıyla karşılaştırın.

`long_breaks` değişkeni, "Günde ortalama kaç dakika mola veriyorsunuz?" sorusuna verilen 3.000 yanıtı temsil eder ve bu yanıtlar daha uzun mola sürelerine sahip olacaktır.

3.000 kişiyi örneklediğimizde, çubuklarımız (bar grafiğindeki frekans değerleri) en yüksek seviyelerine yaklaşık 700–800 kişi civarında ulaşmaktadır.

Şimdi, ortalama olarak yaklaşık 15 dakikalık mola veren 6.000 kişiyi modelleyelim. Bunu simüle etmek için yine Poisson dağılımını kullanacağız, aşağıda gösterildiği gibi:

Böylece, daha uzun molalar veren kişiler için bir dağılımımız ve daha kısa molalar veren kişiler için ayrı bir dağılımımız oldu.

Tekrar dikkat edin: **Ortalama mola süresi olan 15 dakika, dağılımın sol tarafında yer almaktadır** ve en yüksek çubuğun (barın) yaklaşık **1.600 kişi** için olduğunu görebilirsiniz.

`breaks` değişkeni, toplam **9.000 çalışanı** kapsayan bir birleşimdir; hem uzun hem de kısa mola verenleri içerir.

Şimdi, tüm çalışanların dağılımını tek bir görselleştirme içinde görelim:

Grafikte **iki tepe noktası** (hump) olduğunu görebiliriz.

- **Sol tarafta**, yaklaşık **15 dakikalık mola** veren daha büyük bir grup var.
- **Sağ tarafta**, daha uzun molalar veren daha küçük bir grup bulunuyor.

Daha sonra bu grafiği daha ayrıntılı olarak inceleyeceğiz.

Toplam ortalama mola süresini bulmak için şu kodu çalıştırabiliriz:

```
python
KopyalaDüzenle
breaks.mean()
```

Bu kodun çıktısı **29.99 dakika** olup, bu değere **parametre** denir.

Ancak, şirket genelinde ortalama mola süresi **yaklaşık 40 dakika** olarak kabul edilir. Burada, **popülasyonumuz** şirketin **9.000 çalışanının tamamıdır** ve **popülasyon parametremiz 40 dakikadır**.

Gerçek dünyada, genellikle **popülasyon parametresini tahmin etmeye çalışırız**, çünkü her çalışana anket yapacak kaynaklara sahip olmayabiliriz. Bunun yerine, **nokta tahmini** kullanarak popülasyon parametresini belirlemeye çalışırız.

Bunu göstermek için, **rastgele 100 kişiye** molalarının süresini sorduğumuz bir dünya simüle etmek istiyoruz.

Bunu yapmak için, 9.000 çalışan arasından **rastgele 100 çalışan**ın bir örneklemini alacağız:

```
python
KopyalaDüzenle
sample_breaks = np.random.choice(a = breaks, size=100)
# 100 çalışandan oluşan bir örneklem alıyoruz
```

Şimdi, **örneklem ortalaması ile popülasyon ortalaması arasındaki farkı** hesaplayalım:

```
python
KopyalaDüzenle
breaks.mean() - sample_breaks.mean()
```

Çıktı:

```
bash
KopyalaDüzenle
# Ortalamalar arasındaki fark: 0.699 dakika, fena değil!
```

Bu oldukça ilginç çünkü **popülasyonun yalnızca %1'ini (100/9.000) örneklem olarak alarak**, popülasyon parametresine **1 dakikadan daha yakın bir tahmin** elde edebildik. Oldukça iyi bir sonuç!

Şimdiye kadar **ortalama için nokta tahmini** yaptık, ancak **oran (proportion) parametreleri** için de aynı şeyi yapabiliriz.

Oran ile, **iki nicel değer**in oranını kastediyoruz.

Örneğin, **10.000 çalışandan oluşan bir şirketteki etnik dağılım** şu şekildedir:

- **%20** Beyaz (2.000 kişi)
- **%10** Siyah (1.000 kişi)
- **%10** Hispanik (1.000 kişi)
- **%30** Asyalı (3.000 kişi)
- **%30** Diğer (3.000 kişi)

Bu dağılımı şu Python kodu ile tanımlayabiliriz:

```
python
KopyalaDüzenle
employee_races = (["white"]*2000) + (["black"]*1000) + \
                  (["hispanic"]*1000) + (["asian"]*3000) + \
                  (["other"]*3000)
```

Burada, `employee_races` değişkeni şirketin **10.000 çalışanını** temsil eder.

Şimdi, **rastgele 1.000 kişilik bir örneklem alalım** ve her grubun oranlarını hesaplayalım:

```
python
KopyalaDüzenle
import random

demo_sample = random.sample(employee_races, 1000) # 1.000 kişilik örneklem
al
for race in set(demo_sample):
    print(race + " proportion estimate:")
    print(demo_sample.count(race) / 1000.)
```

Bu kodu çalıştırdığımızda aşağıdaki gibi bir çıktı alabiliriz:

```
yaml
KopyalaDüzenle
hispanic proportion estimate: 0.103
white proportion estimate: 0.192
other proportion estimate: 0.288
```

Bu, popülasyona oldukça yakın değerlerdir ve **1.000 kişilik bir örneklemle** 10.000 kişilik şirketin etnik dağılımına dair oldukça iyi bir tahmin elde edebildiğimizi gösterir!