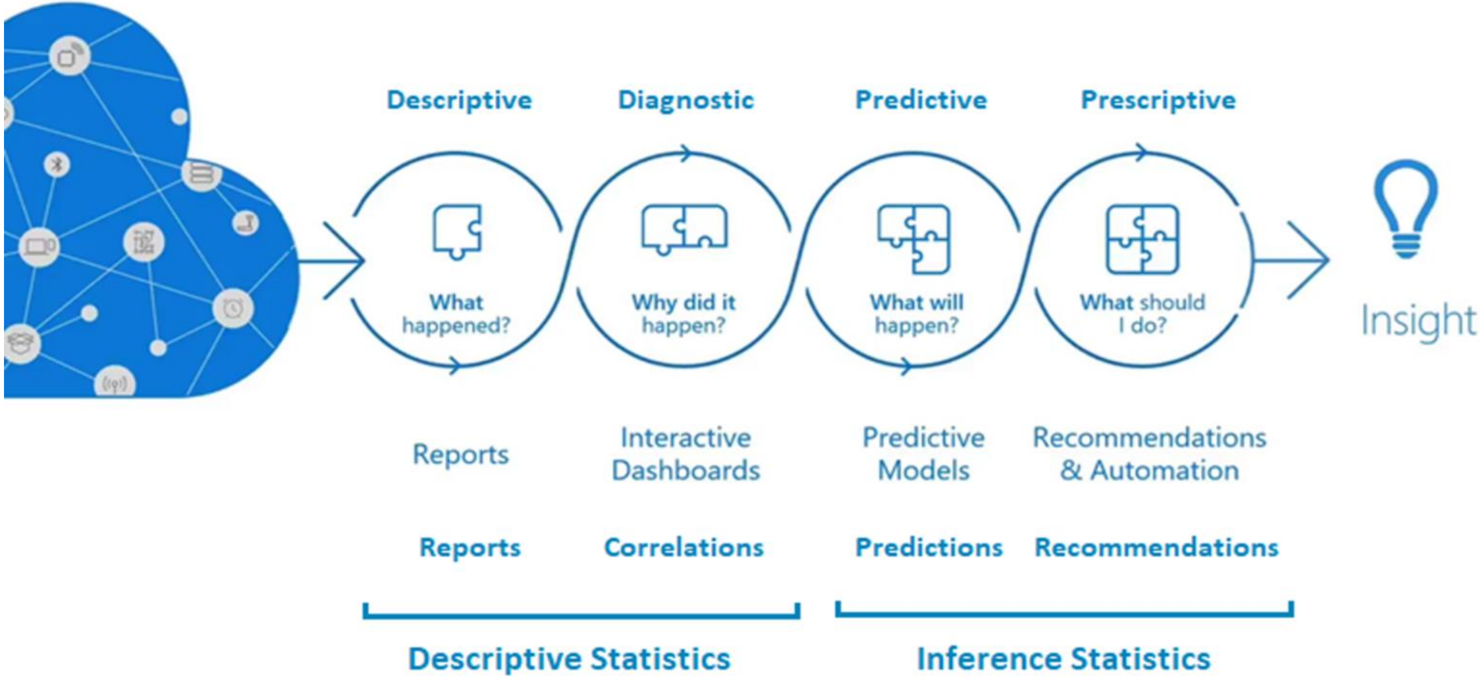
Temel İstatistik ve NumPy

Hafta 4

Betimsel ve Çıkarımsal İstatistik



Önce İstatistik ve Veri Okur Yazarlığı

Veri ile konuşmak istiyorsak İstatistiği iyi bilmek durumundayız.

Önce mod, medyan, aritmetik ortalama, standart sapma (merkezi eğilim ölçüleri) gibi konuları X şirketinin **maaş** verilerine dayanarak göreceğiz.

X şirketinin 10 adet çalışanı var ve çalışanların maaşlarının sıralanmış hali şu şekilde verilmektedir.

2300, 2300, 2300, 2300, 2300, 3800, 4300, 9000, 12000, 16000

Aritmetik Ortalama

Aritmetik ortalama, bir sayı serisindeki sayıların toplamının serinin eleman sayısına (sayı adedine) bölünmesi sonucu elde edilen değerdir.

X şirketindeki maaşların ortalamasını bulmak istiyoruz.

2300+2300+2300+2300+2300+3800+4300+9000+12000+ 16000 = 56.600

56.600 / 10 = 5.660

5.660 Türk Lirası şirkette ortalama maaş değeridir. Bu bilgi şirketin maaş politikası hakkında fikir sahibi olmamız için **yeterli değildir**.

Mod (Tepe Değer)

İstatistik bilimi için mod, sayısal bir veri serisi içinde en sık görülen değerdir. Tepe değer olarak da adlandırılır. Bu sayının tekrar adedine de **frekans** denir.

2300, 2300, 2300, 2300, 2300, 3800, 4300, 9000, 12000, 16000

maaşlar dizisinde en çok tekrar eden maaş (mod değer) **2.300** Türk Lirası.

Mod, maaşların ortalamasının yarısından çok daha az. Üstelik frekansı da çalışan personel sayısının yarısı kadar, yani mod değeri tam **5** kez tekrar etmiş.

Medyan (Ortanca)

Medyan, sayısal bir veri serisi sıralandığında ortada kalan sayıdır. Eleman sayısı çift sayı olan bir seride medyan değeri hesaplanması şu şekildedir.

X Şirketi Maaş Verileri:

2300, 2300, 2300, 2300, 2300, 3800, 4300, 9000, 12000, 16000

serisi 10 personelin maaşını barındırdığı için ortada tek bir sayı yoktur, bu yüzden iki medyanın ortalaması alınır. **(2300 + 3800) / 2 = 3050** Türk Lirası.

Not : *Eleman sayısı tek olan veri serilerinde medyan değeri ortadaki sayıdır.*

Medyan (Ortanca)

Niçin medyan değerine ihtiyaç duyarız ?

Günlük hayattaki veriler dengeli bir dağılım sergileyemeyebilir. Bu verilerin ortalamalarına bakmak çok yanıltıcıdır.

X Şirketi Maaş Ortalaması : **5.660**

X Şirketi Maaş Medyanı : **3.050**

İki değer arasında neredeyse 2 katı fark var. Dağılımın dengeli olup olmadığını teyit etmek için başka değerlere bakmaya devam edelim.

Standart Sapma

Standart sapma, bir serideki sayıların, serinin aritmetik ortalamasından farklarının karelerinin toplamının dizinin eleman sayısının bir eksiğine bölümünün kareköküdür.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

σ : Standart Sapma

N : Dizinin Eleman Sayısı

x_i : Dizinin x. elemanı.

$\bar{x}$: Dizideki Sayıların Aritmetik Ortalaması

$(x_i - \bar{x})^2$: i. elemanın ortalamadan farkının karesini al.

Standart Sapma

Standart sapma ile verilerin ne kadarının **ortalamaya yakın** olduğunu buluruz.

Eğer standart sapma **küçükse** veriler **ortalamaya yakın** yerlerde dağılmışlardır. Bunun tersi olarak standart sapma **büyükse** veriler **ortalamadan uzak** yerlerde dağılmışlardır.

Not : *Bütün değerler aynı olursa standart sapma* ***sıfır*** *olur.*

Standart Sapmayı Yorumlamak

X Şirket Maaşları : **2300, 2300, 2300, 2300, 2300, 3800, 4300, 9000, 12000, 16000**

Y Şirket Maaşları : **5100, 5200, 5300, 5400, 5500, 5600, 5700, 5800, 5900, 6000**

Y şirketinin maaş dağılımının daha düzgün olduğu açıkça görülmektedir. Herhangi bir ölçüm için kullanılan bu dizinin standart sapması, elemanların çoğu ortalamaya yakın olduğundan küçük çıkacaktır.

Y şirketi standart sapması sadece **302,75** ‘dir.

X şirketinde ise maaşların çoğunun ortalamadan uzak değerler aldığı görülüyor. Bu dizinin standart sapması büyük çıkacaktır.

X şirketi standart sapması **4.944,40**‘dır.

Veri Okur Yazarlığı

X Şirketi Maaş Politikası		**Y Şirketi Maaş Politikası**	
Maaş Ortalama :	5.660 TL	Maaş Ortalama :	5.550 TL
Maaş Medyan :	3.050 TL	Maaş Medyan :	5.550 TL
Maaş Mod :	2.300 TL	Maaş Mod :	Yok
Maaş Std. :	**4.944 TL**	Maaş Std. :	**302 TL**
Maks. Maaş :	16.000 TL	Maks. Maaş :	5.100
Min. Maaş :	2.300 TL	Min. Maaş :	6.000
Değişim Aralığı :	13.700	Değişim Aralığı :	900

Standart Sapma ve Ortalama

X ve Y şirketlerinin maaş ortalamasına baksaydık bizim için çok yanıltıcı olacaktı. Bu yüzden **dağılım ölçülerine** bakmak zorundayız.

Maaşların ortalaması ve standart sapmaları arasındaki fark dağılımın ne kadar homojen ve heterojen olduğu hakkında bizi fikir sahibi yaptı.

Standart sapma, ortalamaya göre düşükse verilerin ayırt ediciliği düşüktür ve grup homojendir.

Standart sapma, ortalamaya göre yüksekse grubun heterojen olduğunu ve maaşların ayırt ediciliğinin yüksek olduğunu söyleyebiliriz.

Varyans

Teknik olarak varyans, standart sapmanın karesidir veya diğer deyişle varyans, standart sapmanın karekök alınmamış halidir.

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2$$

Peki o zaman neden varyansa ihtiyacımız var ?

Varyans, standart sapmanın **birimsiz** halidir. X ve Y şirketlerinin standart sapmalarını birbiri arasında mukayese ettik, çünkü aynı dünyanın verileriydi. Yani iki maaş dağılımı da TL cinsinden ve aynı değeri (maaşı) temsil ediyordu.

Farklı dünyadan verilerin ne kadar yayvan olduğunu karşılaştıracaksak varyans değerine ihtiyaç duyarız. Kısacası varyans, dağılımın ne kadar değişkenlik içerdiğini, yayvanlığını ifade eder.

Kovaryans

Kovaryans, değişkenlerin birlikte nasıl değiştiğini araştırmaya ve ölçmeye çalışır.

Bu kavramda, her iki değişken de herhangi bir ilişki belirtmeden aynı şekilde değişebilir.

Kovaryans **iki değişken arasındaki <u>yönü</u>** belirler. **+**, **–** ya da **0** olabilir.

Kovaryans, iki veya daha fazla rasgele değişken kümesi arasındaki korelasyonun güç veya zayıflığının bir ölçümüdür.

$$\sigma_{xy} = \frac{1}{N\text{-}1} \sum_{i=1}^{N} ((X_i - \mu_x) * (Y_i - \mu_y))$$

Peki korelasyon nedir?

Korelasyon

İki veya daha fazla değişken arasındaki ilişkinin varlığı, bu ilişkinin yönü ve şiddeti korelasyon analizi ile belirlenir.

Dikkat, kovaryans sadece ilişkinin **varlığını** ve **yönünü** veriyordu.

İki değişken arasındaki ilişkinin **yönü** ve **derecesi** korelasyon katsayısı ile ifade edilir.

İncelenen değişken sayısı iki tane ise **korelasyon katsayısı** incelenir. İkiden fazla ise **çoklu** veya **kısmi korelasyon katsayısı** incelenir.

Not : *Sözü edilen ilişkinin fonksiyonel şekli ise regresyon analizinin konusunu oluşturur.*

$$r = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 \sum (y-\bar{y})^2}}$$

Korelasyon Katsayısı

Korelasyon katsayısı **-1 < r < 1** aralığında değer alır.

Hesaplanan katsayının aldığı değere bağlı olarak, değişkenler arasında;

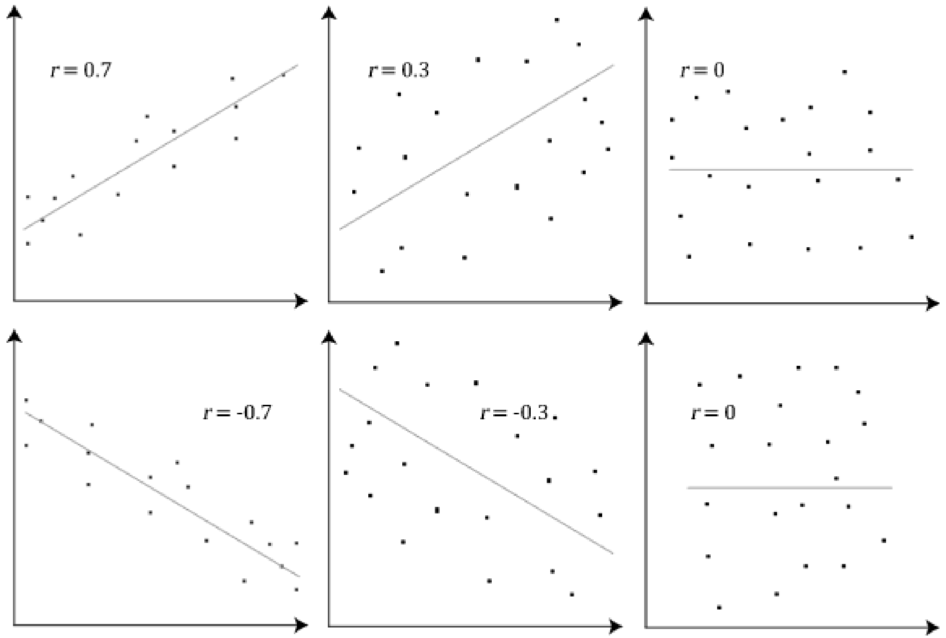
r = -1 (ters yönlü mükemmel ilişki)

r = 0 (ilişki yok.)

r = 1 (aynı yönlü mükemmel ilişki)

r = 0.7 (aynı yönlü iyi ilişki)

r = - 0.3 (ters yönlü orta derece ilişki)



Korelasyon Katsayısı

X Şirketi Maaş (TL cinsinden) Verileri:

2300, 2300, 2300, 2300, 2300, 3800, 4300, 9000, 12000, 16000

X Şirketi Tecrübe (ay cinsinden) Verileri:

6, 12, 12, 18, 18, 32, 48, 64, 80, 120

Yukarıdaki iki dağılımı karşılaştırdığımızda tahmin edebileceğimiz gibi maaş değişkeni ile tecrübe değişkeni arasında güçlü ve pozitif bir ilişki vardır.

Tecrübe arttıkça maaş da artmıştır. Korelasyon katsayısı **pozitif** ve **yüksektir**.

Veri Türleri

Veri dediğimiz şey temelde kategorik veya sayısaldır.

Kategorik değişkenler nelerdir?

Cinsiyet, medeni hali, dini inancı, şehir, ülke, marka, göz rengi, departman, dersi geçme harf notu (AA, CB) vb. değişkenlerdir.

Sayısal değişkenler nelerdir?

Maaş, yaş, boy, dersi geçme puanı, ürünün fiyatı, hava sıcaklığı, kredi miktarı vb.

Burada anlaşılması güç olan konu kategorik ve sayısal olan veriler değil bu verilerin sahip olduğu **ölçek türleridir**.

Ölçek Türleri Nelerdir?

Sayısal değişkenler için ölçek türleri

- Aralık ölçek türü (Oranlanamayan)
- Oran ölçek türü (Oranlanabilen)

Kategorik değişkenler için ölçek türleri

- Nominal ölçek türü (Sınıflandırılabilen)
- Ordinal ölçek türü (Sıralanabilen)

Oran Ölçek Türü ? | Sayısal Değişkenler

Oran ölçek türü **başlangıç noktasını sıfır kabul eder**.

Kilo, mesafe uzunluğu, boy uzunluğu ve yaş buna en belirgin örneklerdir. Hiç negatif bir uzunluk veya negatif bir yaş, negatif kilo duymayız.

Ayrıca bu ölçek türünde **orandan bahsetmek söz konusudur**.

80 kilogram birisi **40** kilogram birisinin iki katı ağırlığındadır. **1000** metre uzunluğundaki yol **4000** metre uzunluğundaki yolun dörtte biridir. **15** yaşındaki bir kişi **5** yaşındaki kişinin **3** misli yaşamıştır.

Özetle, **sıfır** başlangıç noktasıdır ve **oranlayabilmek** söz konusudur.

Aralık Ölçek Türü ? | Sayısal Değişkenler

Başlangıç noktasını sıfır kabul etmez, yani **negatif** ve **pozitif sayılar** söz konusudur ayrıca bu verileri oranlayabilmek de mümkün değildir.

Hava sıcaklığı buna güzel bir örnek. **-5** derece de olabilir **+ 7** derece de.

Ayrıca "**10** derece **2** derecenin **5** katı kadar sıcaktır" demek teknik olarak mümkün değildir.. İki veri arasında oransal bir ilişki yoktur.

Bu ölçekte sıfır noktası yokluğu ifade etmez. "Hava yarın 0 (sıfır) derece olacak" dediğimizde sıfır değeri yokluk ifade etmemektedir.

Ordinal Ölçek Türü ? | Kategorik Değişkenler

Sıralanabilir ölçek türüdür. Hiyerarşik bir düzlemde ifade edilebilen kategorik verileri temsil ederler.

En belirgin örneği askeri sistemidir. **Onbaşı, Yüzbaşı, Binbaşı, Albay** gibi kategorik veriler birbirinden bağımsız değildir, bir **sıralama** söz konusudur.

Eğitim durumu buna güzel bir örnek. **İlkokul < Ortaokul < Lise < Lisans ...** Görüldüğü üzere bu veriler arasında bir ilişki söz konusudur.

Junior, Senior, Expert seviye yazılım geliştiricileri veya dersi geçme harf notu (**AA, BA, BB, CB...**), bankanın müşterilerine atadığı **kredibilite notu(A,B,C...)** buna örnek verilebilir.

Nominal Ölçek Türü ? | Kategorik Değişkenler

Sınıflandırılabilen veri türleridir, aralarında bir **sıralama söz konusu değildir**.

Adana ve İzmir verileri kategoriktir, şehir isimlerini temsil eder ve birbirleri arasında bir sıralama söz konusu olamaz. "**35**" plaka kodu "**01**" plaka kodunun 35 katı değildir. Bir plaka kodu sayısal anlamda hiçbir şey ifade etmez. Çünkü **sınıflandırma amaçlı** tasarlanmıştır.

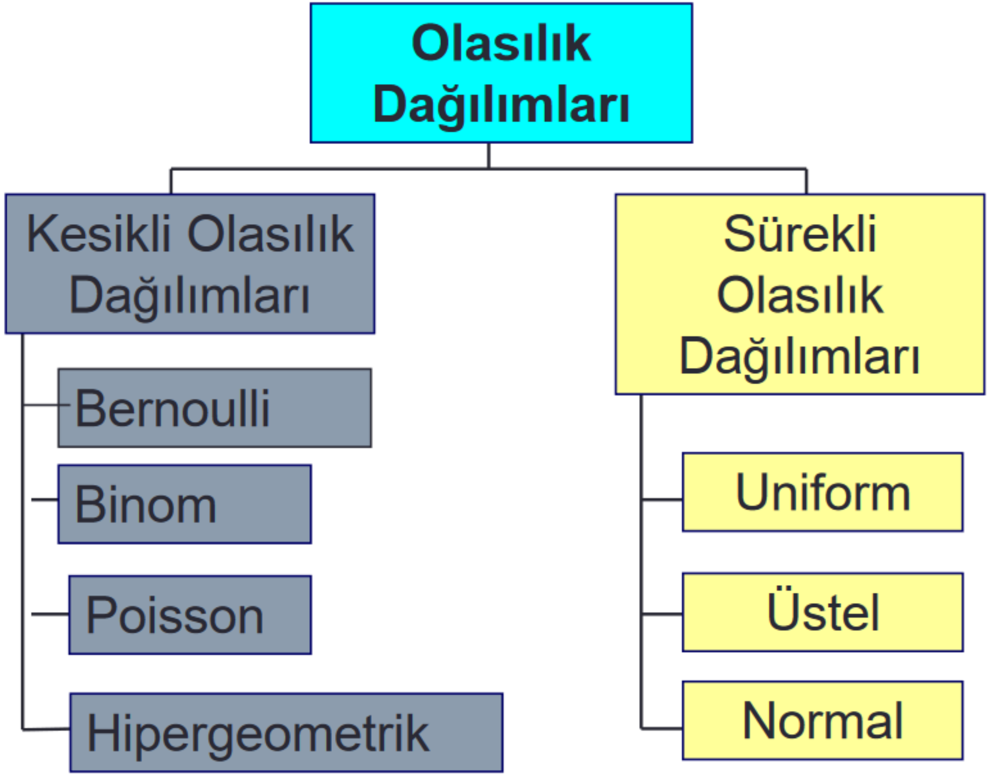
Erkek ve **kadın** bu ölçek türüne örnek verilebilir, birbirlerine üstünlükleri yoktur. **Medeni hali** (evli, bekar), **dini inancı** (musevi, müslüman...) gibi veri türleri de nominal veri türleridir.

Olasılık Dağılımları

Dağılım dediğimiz herhangi bir olay hakkında tutulan gözlem değerlerinin sayısal olarak dökümüdür.

Bu gözlem değerlerinin olasılık olarak gerçekleşme durumlarına da **olasılık dağılımı** denir.

Önce **sürekli** ve **kesikli** kelimelerinin ne anlam ifade ettiğine bakalım.



Sürekli Veriler

Boyunuzun uzunluğu sürekli veridir. (**172 cm ile 173 cm arasında sonsuz sayı var**)

Vücut ağırlığınız sürekli veriye örnektir. (**82.5689 gram**)

Dizüstü bilgisayarınızın ağırlığı da sürekli verilerdir. (**3892,415665 gram**)

Net bir tam sayıyı ifade etmeyebilirler. Bunu 0 ile 1 arasında **sonsuz sayı** olması gibi düşünebiliriz.

Önemli Not : *İki ölçüm arasında sonsuz sayıda anlamlı değerler varsa sürekli veri olduğunu buradan teyit edebiliriz.*

Kesikli Veriler

TR standartlarında ayakkabı numaranızı **kesikli veri** olarak düşünebilirsiniz. **40** veya **41** olabilir ancak **40.5** veya **40.289** olamaz.

Madende çalışan işçilerin akciğerlerindeki leke sayısı **3** veya **4** olabilir ancak **3.5** olamaz. **3.147** olamaz.

İçtiğiniz sigara sayısı da kesikli veridir. Günde **3** veya **18** adet içiyor olabilirsiniz.

Gün içerisinde kaç kez mutfağa girdiğiniz veya kaç kez buzdolabının kapağını açtığınız da kesikli verilere örnek verilebilir.

Önemli Not : *İki ölçüm arasında sonsuz sayıda anlamlı değer yoksa kesikli veri olduğunu buradan teyit edebiliriz.*

Kesikli Olasılık Dağılımıları

Sayılabilir veya sonlu değişkenlerin gerçekleşme durumlarının olduğu dağılım, kesikli olasılık dağılımıdır.

Örneğin; bir zar atılma deneyini incelersek, *“5 sayısının gelme olasılığı 1/6”* olacaktır. Bu dağılımda olasılığını araştırdığımız değişkenlerin tam sayı olması gerekmektedir.

Mesela zarın “**1.7 gelme olasılığına**” bakamayız.

Aynı şekilde para atma deneyi de (yazı veya tura) bir kesikli olasılık dağılımıdır.

Sürekli Olasılık Dağılımıları

Bu dağılımda, bir aralıktaki sonsuz sayıda değişkenin olasılıkları mevcuttur.

Genel olarak ölçüm yoluyla elde edilen değişkenlerin olasılıkları incelenir.

Örnek vermek gerekirse; kişilerin boy uzunlukları, kişilerin ağırlıkları gibi değerler bu dağılımda yer alır.

Kısacası sürekli olasılık dağılımındaki **değişkenler süreklidir** ve ondalıklı değerler alabilir.

Normal Dağılım | Sürekli Olasılık Dağılımı

18. ve 19. yüzyıl matematikçileri, şans oyunlarını ve astronomi gibi diğer bilimsel alanları yöneten kalıpları ve matematiksel modelleri anlamaya hevesliydi.

Bu merak, olasılık teorisinde inanılmaz ilerlemelere izin verdi ve normal dağılım konusunu ortaya çıkardı.

Normal Dağılım | Sürekli Olasılık Dağılımı

~1733
Abraham de Moivre
Derived the formula that was later used to explain the curve of the normal distribution

1805
Adrien-Marie Legendre
Discovered the method of Least Squares in an attempt to minimize the total error in astronomical observational data

1809
Johann Carl Friedrich Gauss
-Used the Least Squares method to fit a probability distribution assuming that the underlying data followed a bell curve
-Published the precise integral to compute the normalizing constant of the normal distribution (Gaussian integral)

1810
Pierre-Simon Laplace
-Used Fourier transform to prove that taking the average of many observations is described by a bell curve (The Central Limit Theorem)
-Validated Gauss's assumption of bell curve shape for the Least Squared method

1835
Adolphe Quetelet
-Discovered that empirical data in social sciences conformed to mathematical law (social science phenomena, when measured in mass, followed the bell curve shape)

Normal Dağılım | Sürekli Olasılık Dağılımı

Merkezi Limit Teoremi Laplace tarafından 1810'da yayınlanmıştır ve Normal Dağılımın bizim işimizle bu kadar alakalı olmasının ana nedenlerinden biridir.

Bir deneyin popülasyonunun birkaç farklı olasılık dağılımı olabilir (normal, çarpık, üstel vb.).

Merkezi Limit Teoremi, popülasyonun dağılımına bakılmaksızın (dağılım sizin için bilinmese bile), yeterince sayıda aynı boyutta rastgele örneklem alındığında ve her örneklem grubunun ortalamasının normal dağılıma yaklaştığını ifade eder.

Normal Dağılımın Özellikleri

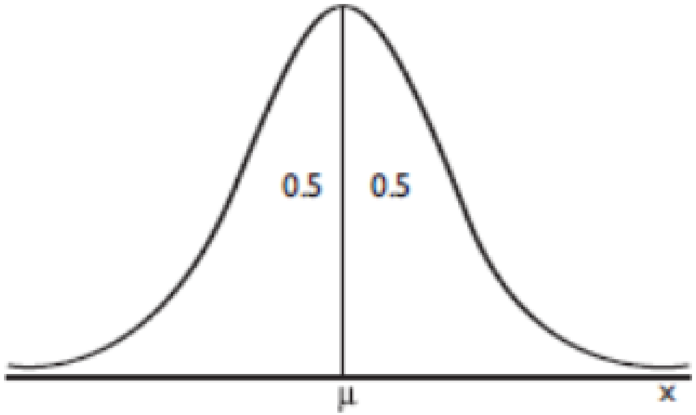
Aritmetik ortalama, mod ve *medyan* birbirine eşittir.

Eğrinin maksimum noktası aritmetik ortalamadır (dolayısıyla mod ve medyandır).

Eğri aritmetik ortalamaya göre ***simetriktir.***

Aritmetik ortalamanın iki yanında kalan alanların değeri **0,5 dir.** Elbette toplam alan 1 dir.

Olasılık değeri 1’den büyük olamaz.

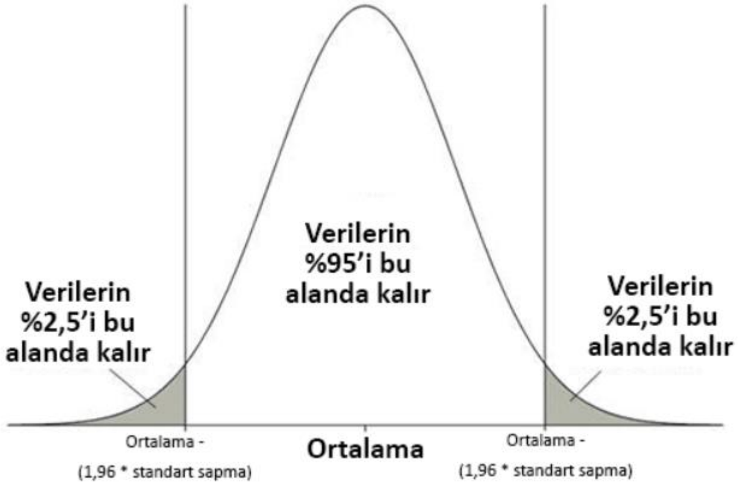


Normal Dağılım | Sürekli Olasılık Dağılımı

Sürekli olasılık dağılımları içinde en önemlisi **normal dağılımdır**. Çünkü günlük yaşamda gözlenen olaylar bu dağılıma uyar.

Bir yatırımın aylık getirileri, üretilen ürünlerin ağırlıkları, boy uzunlukları ve IQ testi sonuçları gibi günlük yaşamdan bir çok örnek bu dağılıma örnek olarak verilebilir.

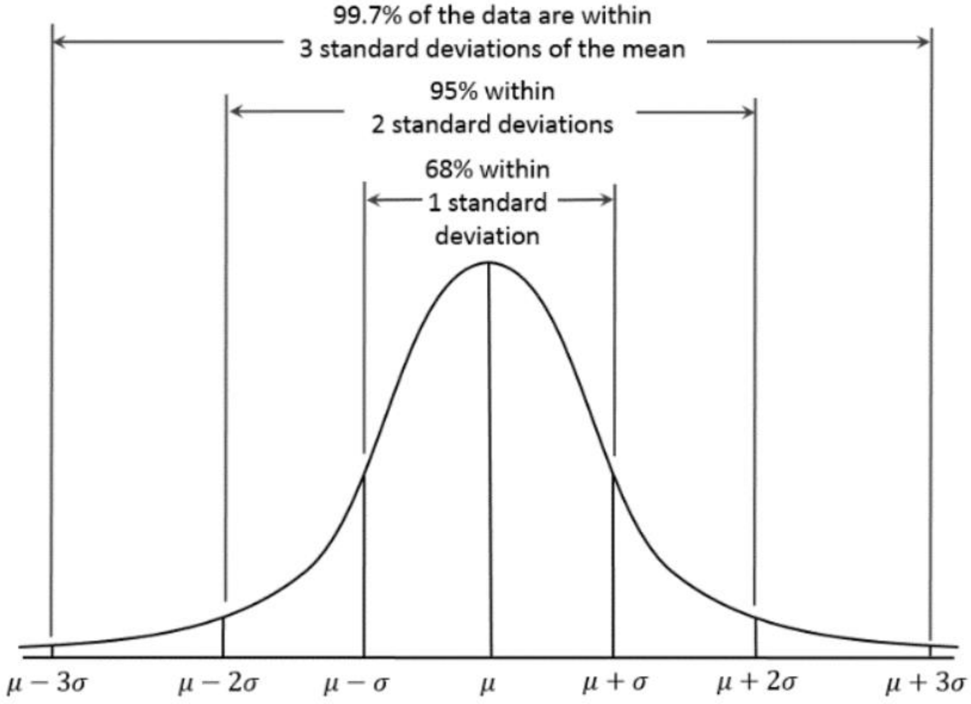
Ayrıca istatistik çıkarsamalarında da temel dağılım olarak kullanılır.



Normal Dağılım | Sürekli Olasılık Dağılımı

Ampirik Kural (% 68-95-99.7 kuralı), normal bir dağılımda, verilerin neredeyse tamamının ortalamanın 3 standart sapması içinde olduğunu belirtir.

Bu yaklaşım, verilerinizdeki aykırı değerleri belirlemeye çalışırken veya dağılımın normalliğini kontrol ederken oldukça kullanışlıdır.



NumPy

NumPy (Numerical Python) bilimsel hesaplamaları hızlı bir şekilde yapmamızı sağlayan bir matematik kütüphanesidir.

Numpy'ın temelini numpy dizileri oluşturur. Numpy dizileri python listelerine benzer fakat hız ve işlevsellik açısından python listelerinden daha kullanışlıdır.

Ayrıca python listelerinden farklı olarak Numpy dizileri homojen yapıda olmalıdır yani dizi içindeki tüm elemanlar aynı veri tipinden olmalıdır.

Numpy ile çalışmak için kütüphaneyi çekirdeğe dahil edelim.

```
import numpy as np
```

Bazı NumPy Matris Özellikleri

```
a = np.random.randint(0,10, size = (3,6))

print(a)
```

```
[[3 6 1 4 8 4]
 [1 3 3 1 9 0]
 [5 7 1 4 6 5]]
```

```
print("Boyutu ", a.ndim)  # vektör olduğu için 1 boyutludur.
print("Şekli", a.shape) # shape, matrisin kaç satır ve sütundan oluştuğunu belirtir.
print("Eleman sayısı", a.size)  # eleman sayısını verir.
print("Değişken tipi ", a.dtype) # matrisin değişken tipini verir.
```

```
Boyutu  2
Şekli (3, 6)
Eleman sayısı 18
Değişken tipi  int32
```

NumPy ile Matris Boyutu Değiştirmek

Bir önceki slaytta oluşturulan **18** gözlem barındıran **a** matrisinin boyutlarını değiştirmek için **reshape( )** fonksiyonu kullanılabilir.

İlk parametre **satır sayısını**, ikinci parametre **sütun sayısını** temsil eder.

```
a.reshape((6,3)) #  6'a 3 yaptık.
```

```
array([[6, 1, 2],
       [0, 3, 7],
       [4, 2, 9],
       [7, 0, 5],
       [0, 3, 7],
       [0, 0, 4]])
```

```
a.reshape((9,2)) #  9'a 2 yaptık.
```

```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12]])
```

```
a.reshape(1,18)
```

```
array([[6, 1, 2, 0, 3, 7, 4, 2, 9, 7, 0, 5, 0, 3, 7, 0, 0, 4]])
```

NumPy ile Matrisleri Birleştirmek

İki matrisin de boyutları uygun olduğu takdirde birleştirme işlemi NumPy’e ait **concatenate( )** fonksiyonu ile mümkündür.

Varsayılan olarak **dikey** birleştirir.

Peki **yatay** birleştirmek istersek ?

```
a = np.array([[1, 2, 3],
              [4, 5, 6]])

b = np.array([[20, 30, 40],
              [50, 60, 70]])

print(a)
print("-" * 15)
print(b)
```

```
[[1 2 3]
 [4 5 6]]
---------------
[[20 30 40]
 [50 60 70]]
```

```
np.concatenate([a,b])
```

```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [20, 30, 40],
       [50, 60, 70]])
```

```
np.concatenate([a,b], axis = 1)
```

```
array([[ 1,  2,  3, 20, 30, 40],
       [ 4,  5,  6, 50, 60, 70]])
```

NumPy ile Diziyi Sıralamak

NumPy hazır fonksiyonları ile dizi sıralanabilir.

```
a = np.array([2, 1, 4, 99, 29,3, 5])
b = np.sort(a)
b
```

```
array([ 1,  2,  3,  4,  5, 29, 99])
```

Listelerde gördüğümüz dilimleme(**slicing**) teknikleriyle NumPy matrisleri tersine çevrilebilir.

```
# Listelerde dilimleme işleminde anlattığımız yöntemler Numpy'de de mevcut.
# Burada yapılan işlem tüm diziyi göster, ancak tersten göster demektir

b[::-1]
```

```
array([99, 29,  5,  4,  3,  2,  1])
```

NumPy ile Transpoze İşlemi

NumPy ile özel bir matris oluşturduk. Transpozunu almak için **reshape( )** veya **.T** fonksiyonları kullanılabilir.

Kısacası NumPy **çok boyutlu matrisler** için tasarlanmış, **çok hızlı işlemler** yapmamızı sağlayan ve matrisler için aklımıza gelebilecek **her fonksiyonu barındıran** bir kütüphanedir.

7 sayılarından oluşan 4 satır 3 sütunluk bir matris oluşturalım.

```
x = np.full((4,3), 7)
x
```

```
array([[7, 7, 7],
       [7, 7, 7],
       [7, 7, 7],
       [7, 7, 7]])
```

x'in transpozunu alalım.

```
x.T  # matrisi devirmiş olduk.
```

```
array([[7, 7, 7, 7],
       [7, 7, 7, 7],
       [7, 7, 7, 7]])
```

Neden Matrisler Bu Kadar Önemli ?

Örneğin, bilgisayarın kamerasına gösterilen rakamları tanıyan bir derin öğrenme modeli geliştireceğimizi düşünelim.

“Bilgisayar bu rakamları nasıl tanır veya öğrenebilir?” sorusunun yanıtı **matrislerdir**.

28 x 28
784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	0	86	253	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	3	162	214	11	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	0	118	253	68	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	0	40	254	98	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	0	112	244	58	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Kaynakça

https://medium.com/@denizkilinc/python-ile-veri-tan%C4%B1maya-ve-temel-i%CC%87statisti%C4%9Fe-dal%C4%B1%C5%9F-7e1028270ac

https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%CC%87%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-2-pandas-dcc12ae01b7d

https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%CC%87%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-1-numpy-750429a0d8e5

https://medium.com/datarunner/veri%CC%87-bi%CC%87li%CC%87mi%CC%87-i%CC%87%C3%A7i%CC%87n-i%CC%87stati%CC%87sti%CC%87k-4c1c72c4158

https://medium.com/analytics-vidhya/the-normal-distribution-for-data-scientists-6de041a01cb9