



ile

# BETİMSEL İSTATİSTİK DERS NOTLARI

# 1. İstatistik Nedir?

Bu ders notları, veri analitiği sürecinin temel aşamalarından biri olan *betimsel istatistik* kavramlarını sistematik ve bütüncül bir bakış açısıyla ele almak amacıyla hazırlanmıştır ve Python ortamında uygulamalı örnekler üzerinden gösterilmektedir. Böylece okuyucuya hem istatistiksel kavramların içeriğini kavrama hem de modern veri bilimi araçlarıyla bu kavramları pratiğe dökme yetkinliği kazandırmak hedeflenmektedir.

İstatistik, verilerin toplanması, düzenlenmesi, analiz edilmesi, yorumlanması ve sonuçların bilimsel karar alma süreçlerine aktarılması ile ilgilenen bilim dalıdır. Temel amacı, gerçek dünyadaki belirsizliklerin sistematik biçimde modellenmesini ve gözlemlere dayalı çıkarımların güvenilir bir şekilde yapılmasını sağlamaktır. İstatistik, yalnızca nicel verilerin özetlenmesiyle sınırlı olmayıp, aynı zamanda örneklerden yola çıkarak ana kütle hakkında genelleme yapılmasını da mümkün kılar. Bu yönyle hem betimsel (descriptive) hem de çıkarımsal (inferential) bileşenleri içeren bütüncül bir metodolojidir.

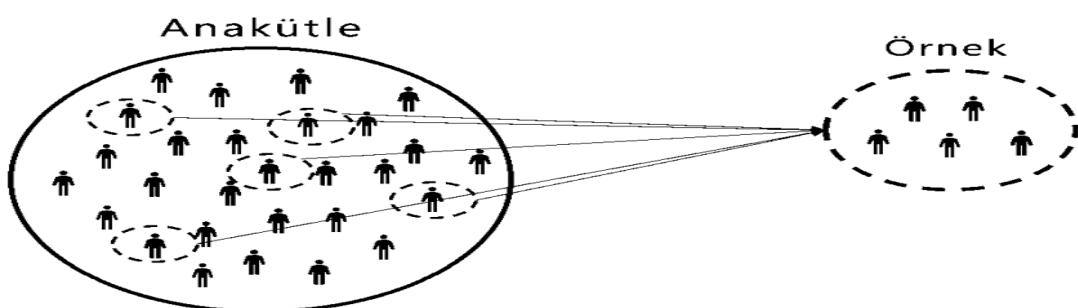
## 2. İstatistiğin Temel Kavramları

### 2.1. Anakütle (Population)

Anakütle, araştırmancın ilgilendiği tüm birimlerin (birey, firma, ürün, olay vb.) oluşturduğu tam kümedir. Araştırmacı çoğu durumda anakütlenin tümüne ulaşamaz; bu nedenle örneklem üzerinde çalışarak anakütle hakkında çıkarım yapılır.

### 2.2. Örneklem (Sample)

Örneklem, anakülteden belirli bir yöntemle seçilen ve anakütleyi temsil eden alt kümedir. Örneklemin temsil gücü, örnekleme yöntemine, örneklem büyüğlüğüne ve örneklem içindeki varyasyona bağlıdır. İstatistiksel çıkarımlar örneklemler üzerinden yapılır.



### **2.3. Değişken (Variable)**

Değişken, gözlem birimlerinin alabileceği farklı değerleri ifade eder. Değişkenler:

Nitel (kategori) değişkenler: Cinsiyet, şehir, marka tercihi gibi kategorilerden oluşur.

Nicel değişkenler: Sayısal değer alır; sürekli (boy, ağırlık) veya kesikli (çocuk sayısı) olabilir.

### **2.4. Veri (Data)**

Veri, değişkenlere ilişkin gözlemlerin topluluğudur. Veri tek başına anlam taşımaz; istatistiksel yöntemlerle işlendiğinde bilgiye dönüşür. Veri türleri, yapılan analiz türünü ve kullanılacak istatistik tekniklerini belirler.

### **2.5. Parametre ve İstatistik**

**Parametre:** Anakütleyi tanımlayan sayısal ölçütür (örneğin anakütle ortalaması  $\mu$ , varyans  $\sigma^2$ ).

**İstatistik:** Örneklemden hesaplanan ölçütür (örneklem ortalaması  $\bar{x}$ , varyans  $s^2$ ).

Çıkarımsal istatistikte amaç, örneklemden elde edilen istatistikleri kullanarak anakütle parametrelerini tahmin etmektir.

### **2.6. Değişken Tiplerine Göre Ölçekler**

Ölçek	Farklılık	Sıra	Miktar	Mutlak başlangıç noktası
Nominal	Evet	Hayır	Hayır	Hayır
Ordinal	Evet	Evet	Hayır	Hayır
Interval	Evet	Evet	Evet	Hayır
Ratio	Evet	Evet	Evet	Evet

İstatistiksel analizlerde veri ölçüği kritik öneme sahiptir:

**Nominal Ölçek:** Sınıflandırma amaçlıdır; büyülüklük ilişkisi yoktur.

**Ordinal Ölçek:** Sıralama ilişkisi vardır, ancak aralar eşit değildir (ör. memnuniyet düzeyi).

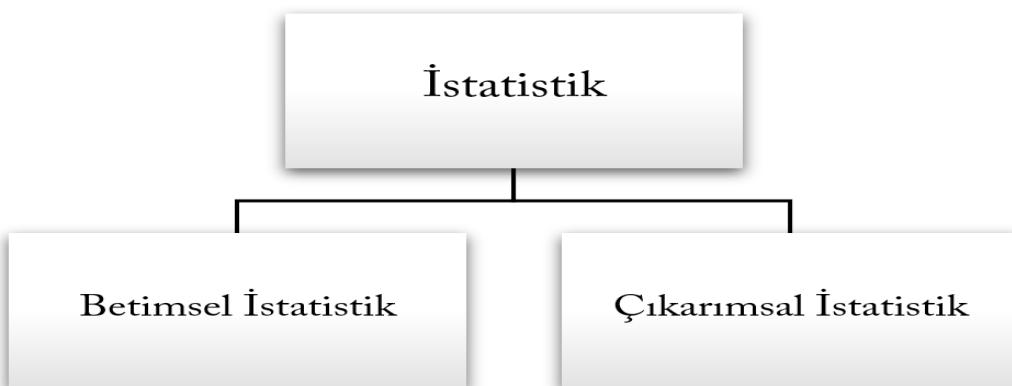
**Aralık ölçek:** Aralıklar eşittir fakat gerçek sıfır noktası yoktur (ör. sıcaklık – Celsius).

**Oran ölçek:** Gerçek sıfır noktası bulunur ve oranlama yapılabilir (gelir, ağırlık).

## 2.7. Değişim ve Dağılım Kavramı

Veriler genellikle tek bir değerde yoğunlaşmaz; farklı gözlemler farklı değerler alır. Bu çeşitliliğe değişim (variability) denir. Değişimin yapısı ve düzeyi, dağılım fonksiyonları, yayılım ölçütleri ve grafikler aracılığıyla incelenir.

## 2.8. Betimsel ve Çıkarımsal İstatistik Ayrımı



**Betimsel istatistik:** Veri setini özetler (ortalama, medyan, mod, standart sapma, grafikler).

**Çıkarımsal istatistik:** Örneklemden hareketle anakütle hakkında tahmin ve hipotez testi yapar.

Bu iki alan birlikte kullanıldığında hem mevcut veriyi anlamak hem de geleceğe yönelik bilimsel sonuçlar üretmek mümkün olur.

# 3. Ölçme Türleri (Measurement Scales)

İstatistiksel analizlerde kullanılan ölçme türleri, değişkenlerin sahip olduğu yapısal özellikleri belirler ve uygulanabilecek istatistik tekniklerini doğrudan etkiler. S. S. Stevens tarafından geliştirilen sınıflandırmaya göre ölçme türleri dört temel kategoride incelenir.

### **3.1. Nominal Ölçek (Nominal Scale)**

Nominal ölçek, gözlem birimlerinin herhangi bir büyüklük ilişkisi olmaksızın sınıflara ayrıldığı ölçüme düzeyidir. Sadece kategorik ayırm vardır.

**Örnek:** Cinsiyet (kadın/erkek), şehir (Ankara/İzmir), marka tercihi.

**İşlem yapılabılırlik:** Sıklık hesaplama, mod, çapraz tablolar, Ki-Kare testi.

Sıralama veya aritmetik işlem yapılamaz.

### **3.2. Ordinal Ölçek (Ordinal Scale)**

Ordinal ölçekte kategoriler arasında mantıklı bir sıralama vardır, ancak kategoriler arasındaki farklar eşit değildir.

**Örnek:** Memnuniyet düzeyi (az–orta–yüksek), eğitim düzeyi, sosyoekonomik statü.

**İşlem yapılabılırlik:** Medyan, yüzde birlikler; non-parametrik testler (Mann-Whitney, Kruskal-Wallis).

Kategoriler arasındaki mesafe ölçülemez.

### **3.3. Aralık Ölçeği (Interval Scale)**

Aralık ölçüği, değerler arasındaki farkların eşit olduğu ancak gerçek bir sıfır noktasının bulunmadığı ölçektir.

**Örnek:** Sıcaklık ( $^{\circ}\text{C}$ ,  $^{\circ}\text{F}$ ), tarih (yıl).

**İşlem yapılabılırlik:** Toplama ve çıkarma yapılabılır; oranlama yapılamaz çünkü sıfır keyfîdir.

Ortalama ve standart sapma hesaplanabilir.

### **3.4. Oran Ölçeği (Ratio Scale)**

Oran ölçüğinde eşit aralıklar vardır ve gerçek sıfır noktası bulunur. Bu nedenle tüm aritmetik işlemler anlamlıdır.

**Örnek:** Ağırlık, gelir, boy, yaş, satış miktarı.

**İşlem yapılabılırlik:** Tüm betimsel ve çıkarımsal istatistiksel teknikler uygulanabilir.

# 4. Veri Yapıları (Data Structures)

Veriler ölçüme türüne, kaynağına ve organizasyon biçimine göre farklı yapıarda sınıflandırılır.

## 4.1. Kesitsel Veri (Cross-Sectional Data)

Belirli bir zaman noktasında birden fazla birimden elde edilen verilerdir.

**Örnek:** 2024 yılında 500 kişilik bir anketin sonuçları.

Analiz türleri: Betimsel istatistik, regresyon, lojistik regresyon, kümeleme.

## 4.2. Zaman Serisi Verisi (Time Series Data)

Aynı birimden farklı zaman noktalarında ölçülen verilerdir.

**Örnek:** Aylık enflasyon oranları, günlük hisse senedi fiyatları.

Analiz türleri: ARIMA, VAR, GARCH, trend ve mevsimsellik analizi.

## 4.3. Panel Veri (Longitudinal Data)

Birden fazla birimden, birden fazla zaman noktasında elde edilen karma veri yapısıdır.

**Örnek:** 2000–2020 arasında 81 il için yıllık işsizlik oranları.

Analiz türleri: Sabit etkiler, rastgele etkiler modelleri (Ekonometride yaygındır).

## 4.4. Büyük Veri (Big Data)

Hacim, çeşitlilik ve hız açısından geleneksel araçlarla işlenmesi güç olan veri türüdür.

**Örnek:** Sosyal medya verisi, IoT sensör verileri.

**Yöntemler:** Dağıtık sistemler, makine öğrenmesi, yapay zekâ teknikleri.

# 5. Değişken Türleri (Types of Variables)

Değişkenler, analizde kullanılacak yöntemlerin belirlenmesi açısından kritik öneme sahiptir.

## 5.1. Nitel (Kategorik) Değişkenler

Sayısal olmayan, kategorilerden oluşan değişkenlerdir.

### a. Nominal değişken

Düzen veya sıralama yoktur.

Örnek: renk, şehir, cinsiyet.

### b. Ordinal değişken

Sıralama vardır ancak aralıklar eşit değildir.

Örnek: eğitim düzeyi, memnuniyet.

## 5.2. Nicel (Sayısal) Değişkenler

### a. Kesikli (Discrete) değişken

Sayıları tam sayı şeklindedir, belirli adımlarla artar.

Örnek: çocuk sayısı, sipariş sayısı.

### b. Sürekli (Continuous) değişken

Belirli bir aralıkta sonsuz sayıda değer alabilir.

Örnek: boy, ağırlık, gelir, sıcaklık.

## 5.3. Bağımlı ve Bağımsız Değişkenler

- Bağımsız değişken (Açıklayıcı değişken)

Modelde sonucu etkilediği düşünülen değişkendir.

- Bağımlı değişken (Açıklanan değişken)

Modelde tahmin edilmek istenen sonuç değişkenidir.

## 6. ORTALAMALAR

Ortalamalar, istatistik ve matematikte bir veri setini veya olasılık dağılımını tek bir representative (temsili) değerle özetlemek için kullanılan temel merkezi eğilim ölçütleridir. Bir veri setindeki tüm değerlerin tipik, merkezi veya beklenen değerini bulmayı amaçlarlar.

Farklı ortalama türleri (aritmetik, geometrik, harmonik vb.) bulunur ve her biri veri setindeki değerlerin yapısına, dağılımın şekline ve ulaşımak istenen sonuca göre farklı bir temsil gücü sunar. Örneğin, değerlerin basit toplamını temsil etmek için aritmetik ortalama kullanılırken, büyümeye oranları ve çarpımsal ilişkiler için geometrik ortalama tercih edilir. Bu nedenle, doğru bağlamda doğru ortalama türünü seçmek, veri analizinin ve yorumlamaların kritik bir adımdır.

Bu ortalama türlerini (Aritmetik, Tartılı, Kareli, Harmonik ve Geometrik Ortalama) Python'da uygulamak için iki temel kütüphaneden faydalanzıız: **NumPy** ve **Pandas**.

### 1. NumPy (Numerical Python)



NumPy, Python'da bilimsel hesaplamaların temel taşıdır.

**Genel Tanım:** NumPy, büyük, çok boyutlu diziler (array) ve matrisler üzerinde hızlı ve verimli işlemler yapmayı sağlayan bir kütüphanedir. Geleneksel Python listelerine göre çok daha yüksek performans sunar, çünkü arka planda C dilinde optimize edilmiş işlemler kullanır.

**Istatistiksel Rolü:** NumPy, ortalamalar, standart sapma, varyans gibi temel istatistiksel fonksiyonlarının çoğunu doğrudan kendi içinde barındırır.

Örneğin, aritmetik ortalamayı hesaplamak için doğrudan `numpy.mean()` fonksiyonunu kullanabiliriz.

Kareli ortalama (RMS) ve harmonik ortalama gibi daha karmaşık ortalamaların hesaplanması da temel matematiksel fonksiyonları (karekök, toplama, bölme) sağlar.

**Kullanım Amacı:** Yüksek performanslı sayısal hesaplamalar ve vektörel işlemler.

## 2. Pandas (Python Data Analysis Library)

Pandas



Pandas, yapılandırılmış verilerle çalışmak için tasarlanmış bir kütüphanedir.

**Genel Tanım:** Pandas, veri manipülasyonu ve analizi için iki ana veri yapısı sunar: Series (tek boyutlu etiketli diziler) ve özellikle büyük veri setlerini tablo formatında tutmak için kullanılan DataFrame (iki boyutlu etiketli tablolar). Pandas, veriyi Excel tabloları, SQL veritabanları veya CSV dosyaları gibi düşünebileceğimiz şekilde organize etmemizi sağlar.

**İstatistiksel Rolü:** Pandas, NumPy üzerine inşa edilmiştir ve veri çerçeveleri üzerinde çalışırken istatistiksel fonksiyonları kolayca uygulamamızı sağlar.

Bir DataFrame sütunundaki verilerin ortalamasını almak gibi toplu işlemleri tek bir komutla (`(df['sütun'].mean())`) gerçekleştirmeyi mümkün kılar.

Özellikle tartılı ortalama gibi, ağırlık (bir sütun) ve değer (başka bir sütun) gerektiren hesaplamalarda veri yönetimini kolaylaştırır.

**Kullanım Amacı:** Veri temizleme, hazırlama, etiketleme ve yapılandırılmış veri analizi.

Ortalama türlerinin Python'da örneklerini yaparken kullanacağımız veri seti aşağıdadır.

Öğrenci	Not	Ağırlık
1	70	2
2	85	1
3	90	3
4	75	2
5	80	1
6	95	2
7	60	1
8	88	2
9	77	1
10	82	1

## 6.1. Aritmetik Ortalama

Aritmetik ortalama, verilerin toplamının gözlem sayısına bölünmesiyle bulunur. En yaygın kullanılan merkezi eğilim ölçüsüdür.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

### KOD

```
import pandas as pd

import numpy as np

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

print(df)

aritmetik_ortalama = df['Not'].mean()

print("Aritmetik Ortalama:", aritmetik_ortalama)
```

The screenshot shows the Spyder IDE interface with a console window titled "Console 1/A". The code in the editor calculates the average grade from an Excel file. The output in the console shows the average grade as 80.2 and the raw data table.

```
Console 1/A ×
Aritmetik Ortalama: 80.2
In [3]: %runfile C:/Users/HP/.spyder-py3/temp.py --wdir
Öğrenci Not Ağırlık
0      1  70      2
1      2  85      1
2      3  90      3
3      4  75      2
4      5  80      1
5      6  95      2
6      7  60      1
7      8  88      2
8      9  77      1
9     10  82      1
Aritmetik Ortalama: 80.2
In [4]:
```

Yukarıda belirtilen kodu çalıştırıldığımızda öğrencilerin not ortalamasının 80.2 olduğu görülür.

## 6.2. Ağırlıklı Ortalama

Her veriye farklı önem (ağırlık) verildiğinde kullanılır. Her veri, kendi ağırlığı ile çarpılır ve toplam ağırlığa bölünür.

$$\bar{x}_w = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$

### KOD

```
import pandas as pd

import numpy as np

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

print(df)

agirlikli_ortalama = np.average(df['Not'], weights=df['Ağırlık'])

print("Ağırlıklı Ortalama:", agirlikli_ortalama)
```

The screenshot shows the Spyder IDE interface with a console window titled "Console 1/A". The code in the editor calculates the weighted average of student grades based on their weights. The output in the console shows the original data frame and the calculated weighted average of 81.875.

```
In [5]: %runfile C:/Users/HP/.spyder-py3/temp.py --wdir
Öğrenci Not Ağırlık
0      1  70      2
1      2  85      1
2      3  90      3
3      4  75      2
4      5  80      1
5      6  95      2
6      7  60      1
7      8  88      2
8      9  77      1
9     10  82      1
Ağırlıklı Ortalama: 81.875

In [6]:
```

Kodu çalıştırduğumızda Ağırlıklı Ortalamanın 81.875 olduğu görülmektedir.

### 63. Kareli Ortalama (Root Mean Square, RMS)

Özellikle değişkenlerin büyüklüklerinin önemli olduğu durumlarda kullanılır. Verilerin karelerinin ortalamasının karekökü alınır.

$$RMS = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}$$

#### KOD

```
import pandas as pd

import numpy as np

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

print(df)

kareli_ortalama = np.sqrt(np.mean(df['Not']**2))

print("Kareli Ortalama (RMS):", kareli_ortalama)
```

```
In [6]: %runfile C:/Users/HP/.spyder-py3/temp.py --wdir
          Öğrenci  Not  Ağırlık
0           1    70      2
1           2    85      1
2           3    90      3
3           4    75      2
4           5    80      1
5           6    95      2
6           7    60      1
7           8    88      2
8           9    77      1
9          10   82      1
Kareli Ortalama (RMS): 80.79108861749543
```

Kod çalıştırıldığında Kareli Ortalamanın 80.79 olduğu görülmektedir.

## 6.4. Harmonik Ortalama

Oran, hız veya yoğunluk gibi ters orantılı büyüklüklerde tercih edilir.

$$H = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

### KOD

```
import pandas as pd

import numpy as np

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

print(df)

harmonik_ortalama = len(df['Not']) / np.sum(1 / df['Not'])

print("Harmonik Ortalama:", harmonik_ortalama)
```

```
Console 1/A ×

In [7]: %runfile C:/Users/HP/.spyder-py3/temp.py --wdir
Öğrenci  Not  Ağırlık
0        1    70      2
1        2    85      1
2        3    90      3
3        4    75      2
4        5    80      1
5        6    95      2
6        7    60      1
7        8    88      2
8        9    77      1
9       10    82      1
Harmonik Ortalama: 78.90566156458821

In [8]:
```

Harmonik ortalamanın ise 78.90 olduğu görülmektedir.

## 6.5. Geometrik Ortalama

Büyüme oranları ve faiz, endeks hesaplamalarında kullanılır. Verilerin çarpımının n'inci kökü alınır.

$$G = \left( \prod_{i=1}^n x_i \right)^{\frac{1}{n}}$$

### KOD

```
import pandas as pd

import numpy as np

from scipy.stats import gmean # geometrik ortalama için

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

print(df)

geometrik_ortalama = gmean(df['Not'])

print("Geometrik Ortalama:", geometrik_ortalama)
```

**NOT:** 3. satırındaki koda bakıldığından;

“Geometrik ortalama hesaplamasında `np.prod` büyük sayı çarpımlarında hata verebileceği için `scipy.stats.gmean` kullanıldı; bu yöntem `overflow` ve `nan` sorunlarını önler.”

The screenshot shows the Spyder Python IDE interface with a console window titled "Console 1/A". The console output is as follows:

```
In [9]: %runfile C:/Users/HP/.spyder-py3/temp.py --wdir
Öğrenci Not Ağırlık
0      1   70      2
1      2   85      1
2      3   90      3
3      4   75      2
4      5   80      1
5      6   95      2
6      7   60      1
7      8   88      2
8      9   77      1
9     10   82      1
Geometrik Ortalama: 79.57145201141812

In [10]:
```

The script runs successfully, displaying the calculated geometric mean of 79.57145201141812.

Öğrencilerin ders notlarının Geometrik Ortalamasının 79.57 olduğu görülmektedir.

### 6.6. Mod (Mode)

Mod, bir veri setinde en sık tekrar eden değeri ifade eder.

Özellikler: Veri setindeki merkezi eğilimi gösterir; nitel ve nicel veriler için kullanılabilir.

Örnek: Sınav notları {70, 75, 80, 80, 85} → Mod = 80

$$\text{Mode} = \operatorname{argmax}_{x_i} f(x_i)$$

Burada  $f(x_i)$ ,  $x_i$  değerinin frekansıdır.

### KOD

```
import pandas as pd

import numpy as np

from scipy import stats

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

notlar = df['Not']

mod = stats.mode(notlar, keepdims=True).mode[0]

print("Mod:", mod)
```

**NOT:** `scipy.stats`, bilimsel hesaplamalar için temel Python kütüphanelerinden biri olan **SciPy**'nin bir parçasıdır ve özellikle istatistiksel çıkarım ve modelleme için hayatı araçları sağlar.

## 6.7. Medyan (Median)

Medyan, veriler küçükten büyüğe sıralandığında ortada kalan değerdir.

Özellikler: Aşırı değerlerden etkilenmez, özellikle çarpık dağılımlarda merkezi eğilimi gösterir.

**Hesaplama:**

Tek sayıda gözlem → ortadaki değer

Çift sayıda gözlem → ortadaki iki değerin aritmetik ortalaması

Veriler  $x_1 \leq x_2 \leq \dots \leq x_n$  şeklinde sıralandığında:

$$\text{Medyan} = \begin{cases} x_{(n+1)/2}, & n \text{ tek} \\ \frac{x_{(n/2)} + x_{(n/2+1)}}{2}, & n \text{ çift} \end{cases}$$

### KOD

```
import pandas as pd
import numpy as np
from scipy import stats
dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav
notları.xlsx"
df = pd.read_excel(dosya_yolu)
notlar = df['Not']
medyan = np.median(notlar)
print("Medyan:", medyan)
```

## 6.8. Kartiller (Quartiles)

Kartiller, veri setini dört eşit parçaya bölen değerlerdir:

Q1 (Alt Çeyrek): Verilerin %25'i bu değerin altındadır

Q2 (Orta Çeyrek / Medyan): %50

Q3 (Üst Çeyrek): %75

**Kullanım:** Dağılımın yayılımını ve aykırı değerleri incelemeye etkilidir

Veri setini dört eşit parçaya böler:

- Alt çeyrek:  $Q_1 = x_{(n+1)/4}$
- Medyan / Ortanca:  $Q_2 = x_{(n+1)/2}$
- Üst çeyrek:  $Q_3 = x_{3(n+1)/4}$

### KOD

```
import pandas as pd

import numpy as np

from scipy import stats

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

notlar = df['Not']

Q1 = np.percentile(notlar, 25)

Q2 = np.percentile(notlar, 50) # Medyan

Q3 = np.percentile(notlar, 75)

print(f"Kartiller: Q1={Q1}, Q2={Q2}, Q3={Q3}")
```

## 6.9. Düzeltilmiş Ortalama (Trimmed / Winsorized Mean)

Aşırı uç değerlerin etkisini azaltmak için bazı gözlemler atılır veya sınırlandırılır.

Trimmed Mean: En uç %5 veya %10 değerler atılır

Winsorized Mean: En uç değerler, en yakın uç değerlerle değiştirilir

Amaç: Ayrıntıları değiştirmek

### KOD

```
import pandas as pd  
  
import numpy as np  
  
from scipy import stats  
  
dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav  
notları.xlsx"  
  
df = pd.read_excel(dosya_yolu)  
  
notlar = df['Not']  
  
trimmed_mean = stats.trim_mean(notlar, 0.1) # uç %10 değerler atıldı  
  
print("Düzeltilmiş Ortalama (Trimmed 10%):", trimmed_mean)
```

## 6.9. Persantiller (Percentiles)

Persantiller, veriyi %'lık dilimlere böler.

Örnek: 90. persantil → verilerin %90'ı bu değerin altında

Kartiller aslında belirli persantillerdir: Q1 = 25. persantil, Q2 = 50., Q3 = 75.

$$P_p = x_k + \left( \frac{np - k}{1} \right) (x_{k+1} - x_k)$$

## KOD

```
import pandas as pd

import numpy as np

from scipy import stats

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

notlar = df['Not']

p10 = np.percentile(notlar, 10)

p90 = np.percentile(notlar, 90)

print(f"Persantiller: 10. persantil={p10}, 90. persantil={
```

## 6.10 Çarpım / Çarşıklık (Skewness)

Veri dağılımının simetri özelliğini ölçer:

0: Simetrik dağılım

Pozitif: Sağa çarpık (sağ kuyruk uzun)

Negatif: Sola çarpık (sol kuyruk uzun)

Kullanım: Merkezi eğilim ve yayılım ölçülerinin yorumlanması sırasında önemli

$$\text{Skewness} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left( \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{3/2}}$$

## KOD

```
import pandas as pd

import numpy as np

from scipy import stats

dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"

df = pd.read_excel(dosya_yolu)

notlar = df['Not']

skewness = stats.skew(notlar)

print("Çarşıklık (Skewness):", skewness)
```

## 6.11. Basıklık (Kurtosis)

Dağılımin sivrilik veya yayvanlık derecesini ölçer:

- 3: Normal dağılımin basıklığı (mesokurtik)
- >3: Sivri dağılım (leptokurtik) → kuyruklar kalın
- <3: Yayvan dağılım (platykurtik) → kuyruklar ince

Amaç: Ayrıntılı değerlerin ve uç değerlerin yoğunluğunu anlamak

$$\text{Kurtosis} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left( \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^2},$$

### KOD

```
import pandas as pd  
  
import numpy as np  
  
from scipy import stats  
  
dosya_yolu = r"C:\Users\HP\Desktop\Kurs Kaynakları (1)\VeriSeti\sınav notları.xlsx"  
  
df = pd.read_excel(dosya_yolu)  
  
notlar = df['Not']  
  
kurtosis = stats.kurtosis(notlar, fisher=False) # Fisher=False -> normal dağılım=3  
  
print("Basıklık (Kurtosis):", kurtosis)
```

```
In [12]: %runfile C:/Users/HP/.spyder-py3/temp.py --wdir  
Mod: 60  
Medyan: 81.0  
Kartiller: Q1=75.5, Q2=81.0, Q3=87.25  
Düzeltilmiş Ortalama (Trimmed 10%): 80.875  
Persantiller: 10. persantil=69.0, 90. persantil=90.5  
Çarpıklık (Skewness): -0.5066371388833639  
Basıklık (Kurtosis): 2.645949981633111
```

Tüm kodları çalıştırıldığımızdaki betimsel istatistik sonuçlarını yukarıda görüyoruz.

Söz konusu öğrenci notları veri setine uygulanan tanımlayıcı istatistiksel analiz, dağılımin merkezi eğilim, yayılım ve şekil parametrelerini ortaya koymuştur. Merkezi eğilim ölçütlerinde, Medyan (81.0) ve aykırı değerlerden arındırılmış Düzeltilmiş Ortalama (80.875) birbirine yakın yüksek bir merkez gösterirken, Mod (60) en sık tekrarlanan tek değeri işaret eder. Dağılımin şekli incelendiğinde, Çarpıklık değerinin {-0.5066} ile negatif olması, notların çoğunlukla yüksek değerlerde yiğildiğini ve kuyruğun düşük notlara doğru uzadığını (sola çarpık) gösterir. Basıklık değeri ise {2.6459} ile )3)'ten düşük çıkmış olup, dağılımin zirvesinin normal dağılıma göre daha basık ve kuyruklarının daha ince olduğunu (Platykurtic) göstermektedir; bu da aşırı uç değerlerin düşük sıklıkta olduğunu ima eder. Kartiller (Q1=75.5, Q3=87.25) ise notların orta %50'lik diliminin nispeten dar bir aralıkta toplandığını ve genel başarının yüksek ve tutarlı olduğunu teyit etmektedir.

## 7. Veri Görselleştirme Teknikleri

Veri görselleştirme, betimsel istatistikte veriyi daha anlaşılır hâle getirmek, dağılım yapısını incelemek, ilişkileri tanımlamak ve olası anormallikleri tespit etmek için kullanılan temel araçlardan biridir. Grafiksel gösterimler, sayısal özet istatistiklerin sağlayamadığı yapısal örüntüleri ortaya çıkarır ve veri analizi sürecinin vazgeçilmez bir bileşenidir.

Veri görselleştirme teknikleri için uygulamalarda kullanılacak veri setinin bilgileri şunlar:

Bu veri seti:

- ✓ 50 gözlem
- ✓ Tarih, yaşı, gelir, harcama
- ✓ Cinsiyet, eğitim, ürün tipi
- ✓ Sağlık skoru
- ✓ Satış adedi, reklam bütçesi
- ✓ Gerçekçi dağılımlar + ilişkili değişkenler

Bu veri setini herhangi bir yapay zeka modeline oluşturtabilirsiniz.

Kütüphaneler:

**Matplotlib:**



Matplotlib, Python'ın temel grafik çizim kütüphanesidir.

Düşük seviyeli (low-level) kontrol sağlar

Her tür grafiği oluşturabilir (çizgi, histogram, scatter, bar)

Esnek ancak biraz daha fazla kod gerektirir

**Seaborn:**



Seaborn, matplotlib tabanlı yüksek seviyeli (high-level) istatistiksel görselleştirme kütüphanesidir.

Varsayılan olarak estetik ve düzenli grafikler

KDE, boxplot, violin plot, heatmap gibi gelişmiş istatistiksel grafikler

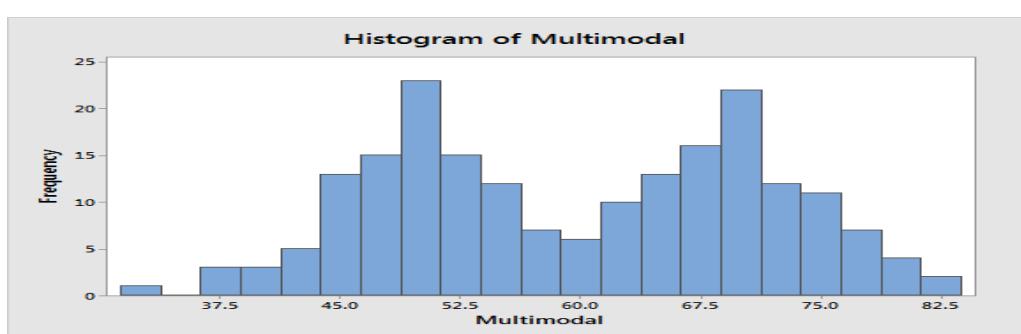
DataFrame yapılarıyla doğrudan uyumlu

## 7.1 Histogram, Kutu (Box) Grafiği ve Yoğunluk Grafiği

### Histogram

Histogram, sürekli veya aralıklı nicel değişkenlerin dağılımını incelemek için kullanılan temel bir grafiktir. Veri belirli sınıf aralıklarına (bins) ayrılır ve her bir aralığın frekansı dikdörtgen sütunlarla gösterilir.

Histogram, dağılımin simetrik, sağ/sol çarpık, tek tepe veya çok tepe gibi özelliklerini incelemede oldukça etkilidir.



## KOD

```
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_excel(r"C:\Users\HP\Desktop\karma_veri_seti.xlsx")

plt.figure(figsize=(7,5))

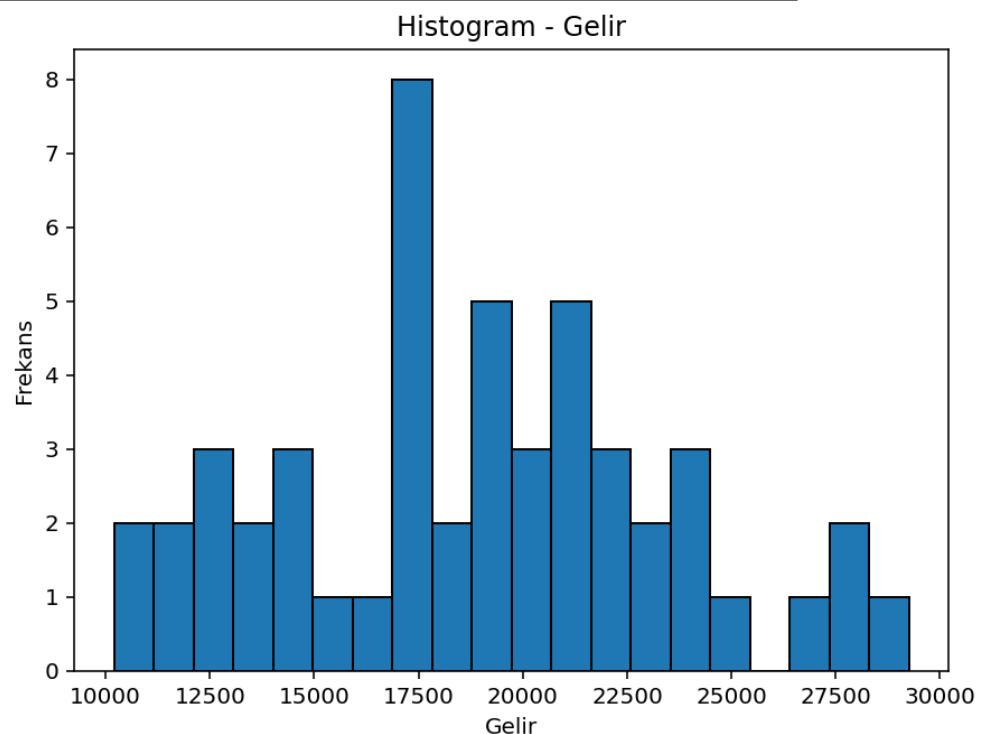
plt.hist(df["Gelir"], bins=20, edgecolor="black")

plt.title("Histogram - Gelir")

plt.xlabel("Gelir")

plt.ylabel("Frekans")

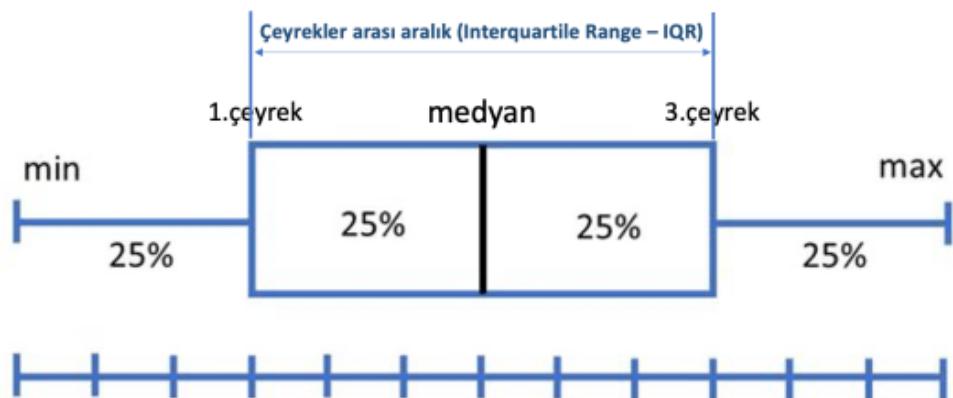
plt.show()
```



Verilen histogram, örneklemdeki gelir dağılımının genel yapısını göstermektedir. Grafik incelendiğinde, gelirlerin yaklaşık olarak 17.000–22.000 TL aralığında yoğunluğu ve bu aralığın dağılımın merkezi eğilimini temsil ettiği görülmektedir. Histogramın sağa çarpık bir yapıya sahip olduğu, yani yüksek gelir değerlerinin daha seyrek ve uç değerler olarak ortaya çıktığı anlaşılmaktadır. Alt gelir dilimlerinde gözlenen düşük frekanslar, gelir dağılımında heterojenliğin ve olası aykırı değerlerin varlığına işaret etmektedir. Bu dağılım, temel

betimleyici istatistikler ile desteklendiğinde, gelir verisinin merkezi eğilim, yayılım ve dağılım şekli hakkında net bir görsel bilgi sunmaktadır.

### Kutu (Box) Grafiği



Kutu grafiği (boxplot), bir değişkenin medyanını, alt ve üst kartillerini (Q1, Q3) ve akyarı değerlerini görselleştiren bir yöntemdir.

Kutu grafiği aşağıdaki bileşenlerden oluşur:

Kutu → Q1 ile Q3 arasındaki aralık

Çizgi → Medyan

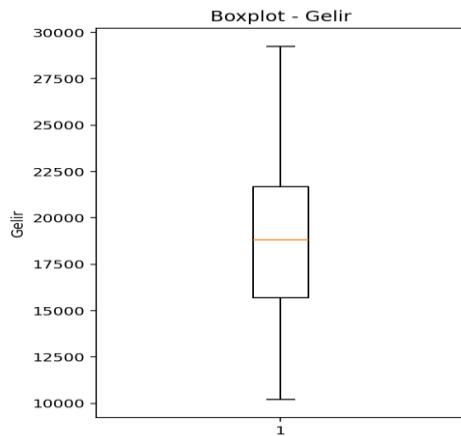
Büyüklar →  $1.5 \times \text{IQR}$  sınırları

Noktalar → Akyarı değerler

Bu görselleştirme özellikle dağılım karşılaştırmalarında ve akyarı değer analizinde kullanılır.

### KOD

```
import pandas as pd  
  
import matplotlib.pyplot as plt  
  
df = pd.read_excel(r"C:\Users\HP\Desktop\karma_veri_seti.xlsx")  
  
plt.figure(figsize=(4,6))  
  
plt.boxplot(df["Gelir"], vert=True)  
  
plt.title("Boxplot - Gelir")  
  
plt.ylabel("Gelir")  
  
plt.show()
```



Bu kutu grafiği, Gelir dağılımının merkezi eğilim ve yayılım ölçülerini göstermektedir. Gelirlerin medyan değeri (ortanca) yaklaşık 18.750 birimdir. Veri setinin orta %50'lük kesimi, birinci çeyrek olan 15.750 birim ile üçüncü çeyrek olan 21.750 birim arasındaki aralıkta yer almaktadır. Bu durum, çeyrekler arası açıklığın yaklaşık 6.000 birim olduğunu gösterir. Bölgeler, minimum değerin yaklaşık 10.000 birim ve maksimum değerin yaklaşık 29.500 birim olduğunu ortaya koymaktadır. Medyanın kutuya yakın konumu ve bölgelerin benzer uzunlukları, dağılımin göreceli olarak simetrik olduğunu ve incelenen aralıkta belirgin bir aykırı değere rastlanmadığını ifade etmektedir.

## 7.2 Korelasyon Matrisi ve İşı Haritası

### Korelasyon Matrisi

Korelasyon matrisi, nicel değişken çiftleri arasındaki doğrusal ilişkilerin gücünü ve yönünü gösteren tablodur. Matrisin elemanları Pearson korelasyon katsayısını içerir ( $r \in [-1, 1]$ ).

$r \approx +1 \rightarrow$  Güçlü pozitif ilişki

$r \approx -1 \rightarrow$  Güçlü negatif ilişki

$r \approx 0 \rightarrow$  İlişki yok veya zayıf

## KOD

```
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_excel(r"C:\Users\HP\Desktop\karma_veri_seti.xlsx")

numeric_df = df.select_dtypes(include=["int64", "float64"])

corr_matrix = numeric_df.corr()

plt.figure(figsize=(8, 5))

plt.imshow(corr_matrix, cmap="coolwarm", interpolation="nearest")

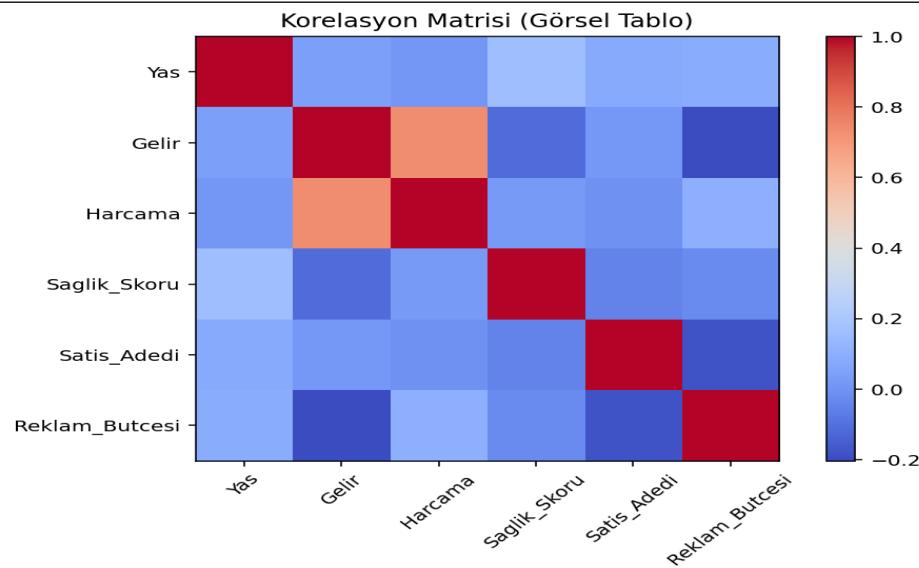
plt.colorbar()

plt.xticks(range(len(corr_matrix.columns)), corr_matrix.columns, rotation=45)

plt.yticks(range(len(corr_matrix.columns)), corr_matrix.columns)

plt.title("Korelasyon Matrisi (Görsel Tablo)")

plt.show()
```



Gelir - Harcama: Güçlü pozitif korelasyon (koyu kırmızı) gözlemlenmektedir. Bu, geliri yüksek olan bireylerin harcama miktarlarının da yüksek olduğunu gösterir.

Sağlık Skoru - Harcama: Orta düzeyde negatif korelasyon (açık mavi/beyaza yakın) mevcuttur. Bu, sağlık skoru arttıkça harcamanın bir miktar azaldığını veya tam tersini işaret edebilir, ancak ilişki zayıftır.

Satış Adedi - Reklam Bütçesi: Güçlü pozitif korelasyon (koyu kırmızı) görülmektedir. Bu, reklam bütçesi arttıkça satış adetlerinin de önemli ölçüde arttığını veya tam tersini gösterir; bu iki değişken arasında güçlü bir doğrusal ilişki vardır.

**Yaş - Diğer Değişkenler:** Yaş değişkeninin, incelenen diğer değişkenlerle (Gelir, Harcama vb.) arasındaki ilişkisi çok zayıf veya yoktur (koyu mavi/beyaza yakın), bu da bu değişkenlerin yaşa bağlı olmadan hareket ettiğini düşündürmektedir.

Genel olarak matris, özellikle ekonomik (Gelir, Harcama, Reklam Bütçesi, Satış Adedi) değişkenler arasında anlamlı pozitif ilişkiler olduğunu, ancak demografik (Yaş) ve sağlık (Sağlık Skoru) değişkenlerinin diğerleriyle daha zayıf bağlantılar sergilediğini göstermektedir.

### **Isı Haritası (Heatmap)**

Isı haritası, korelasyon matrisini renk skaları ile görselleştirmek için ilişkilerin hızlı biçimde yorumlanması sağlar. Koyu renkler güçlü ilişkileri, açık renkler zayıf ilişkileri temsil eder. Çok değişkenli veri analizinde oldukça etkilidir.

### **7.3 Zaman Serisi Grafiklerinin Oluşturulması**

Zaman serisi grafikleri, gözlemlerin zamana bağlı olarak değişimini incelemek için kullanılır.

#### **Temel özellikleri:**

X eksen → Zaman

Y eksen → Değer

Trend, mevsimsellik, yapısal kırılmalar gibi bileşenlerin analiz edilmesi için kullanılır.

Finans, iktisat, mühendislik ve doğa bilimlerinde en yaygın kullanılan grafik türlerinden biridir.

## KOD

```
import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_excel(r"C:\Users\HP\Desktop\karma_veri_seti.xlsx")

df.columns = df.columns.str.strip().str.replace("ş", "s").str.replace("Ş", "S")

tarih_sutun = None

for col in df.columns:

    if "tarih" in col.lower():

        tarih_sutun = col

        break

if tarih_sutun is None:

    raise ValueError("Veri setinde 'Tarih' sütunu bulunamadı!")

satis_sutun = None

for col in df.columns:

    if "satis" in col.lower():

        satis_sutun = col

        break

if satis_sutun is None:

    raise ValueError("Veri setinde 'Satis' sütunu bulunamadı!")

df[tarih_sutun] = pd.to_datetime(df[tarih_sutun])

df = df.sort_values(tarih_sutun)

plt.figure(figsize=(12, 6))

plt.plot(df[tarih_sutun], df[satis_sutun], marker='o')

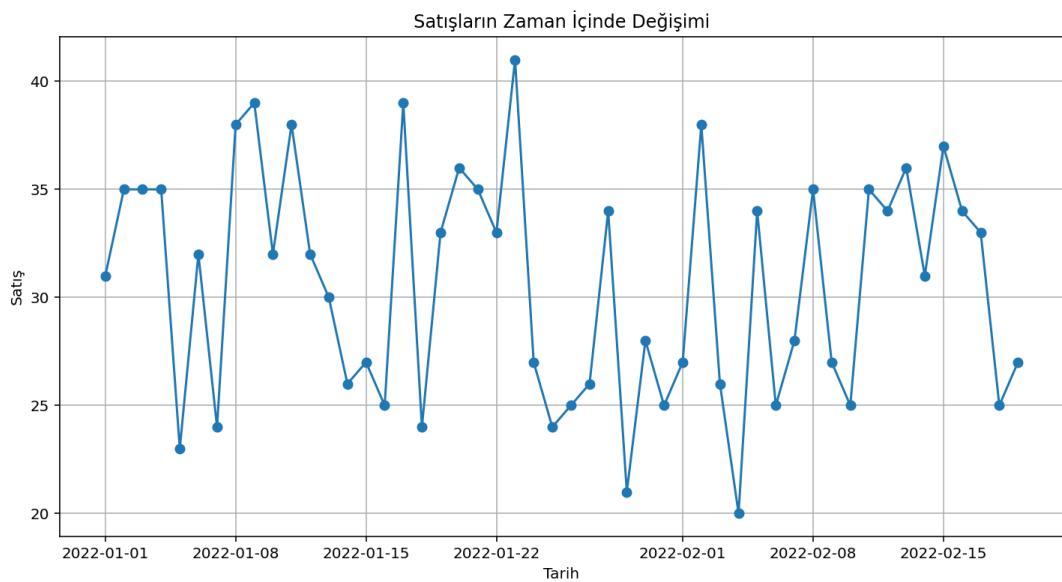
plt.title("Satışların Zaman İçinde Değişimi")

plt.xlabel("Tarih")

plt.ylabel("Satış")

plt.grid(True)

plt.show()
```



Grafik, 2022 Ocak-Şubat döneminde günlük satışların yaklaşık 20 ile 41 birim arasında büyük bir dalgalanma gösterdiğini ortaya koymaktadır. Kısa dönemde belirgin bir uzun vadeli trend (yükseliş veya düşüş) gözlenmezken, satışlar yüksek oynaklık sergilemiş ve medyan seviyesi etrafında salınmıştır. Bu yapı, satış performansının istikrarsız olduğunu ve kısa süreli, periyodik hareketlerin baskın olduğunu işaret etmektedir.

# 8. Değişkenlik

## 1. Değişim Aralığı (Range) ve Kartillerarası Fark (IQR)

Değişim Aralığı (Range)

Bir veri setindeki en büyük değer ile en küçük değer arasındaki farktır.

**Formül:**

$$R = X_{\max} - X_{\min}$$

Verinin toplam yayılımını gösterir ancak uç değerlere duyarlıdır.

Kartillerarası Fark (Interquartile Range, IQR)

Verilerin ortadaki %50'sinin yayılımını gösteren ölçütür.

**Formül:**

$$IQR = Q_3 - Q_1$$

Burada  $Q_1$  veri setinin alt çeyreğini,  $Q_3$  üst çeyreğini temsil eder.

Uç değerlere karşı dayanıklı bir değişkenlik ölçüsüdür.

## 8.1. Ortalama Sapma (Mean Absolute Deviation)

Veri değerlerinin ortalamadan olan mutlak sapmalarının aritmetik ortalamasıdır.

**Formül:**

$$MAD = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

Ortalama etrafındaki tipik sapmanın büyüklüğünü gösterir. Mutlak değer kullanıldığı için negatiflik problemi yoktur, ancak standart sapma kadar yaygın kullanılmaz.

## 8.2. Varyans, Standart Sapma ve Değişim Katsayısı

Varyans (Variance)

Veri değerlerinin ortalamadan olan kareli sapmalarının ortalamasıdır.

**Formül (popülasyon):**

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

**Formül (örneklem):**

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Birimin karesi cinsinden ölçüm yapar.

Standart Sapma (Standard Deviation)

Varyansın kareköküdür; verinin ortalama etrafında tipik olarak ne kadar dağıldığını gösterir.

**Formül:**

$$s = \sqrt{s^2}$$

Birim veri birimiyle aynıdır ve en yaygın değişkenlik ölçüsüdür.

Değişim Katsayısı (Coefficient of Variation, CV)

Standart sapmanın ortalamaya oranıdır; bağıl değişkenlik ölçüsüdür.

Farklı birimlerdeki ya da farklı ortalamalara sahip veri setlerini kıyaslamak için kullanılır.

**Formül:**

$$CV = \frac{s}{\bar{x}} \times 100$$

Yüzde olarak ifade edilir.

### 8.3 Standart Sapmanın Yorumu

Standart sapma, veri setindeki değerlerin ortalama etrafında ne kadar dağıldığını gösterir.

Bilimsel yorum ilkeleri:

Küçük standart sapma → Veriler ortalamaya yakın, daha homojen.

Büyük standart sapma → Veriler ortalamadan daha uzak, heterojenliğin fazla olduğunu gösterir.

Normal dağılıma yakın veri setlerinde:

Değerlerin yaklaşık %68'i ortalamanın  $\pm 1$  standart sapma aralığındadır.

Değerlerin %95'i ortalamanın  $\pm 2$  standart sapma aralığındadır.

Bu nedenle standart sapma, verinin istatistiksel yayılımının en anlamlı özetleyicilerinden biridir.

## KOD

```
import numpy as np

import pandas as pd # Verileri tablo halinde göstermek için

np.random.seed(42) # Sonuçların her çalışmada aynı olması için sabitleme

veri_seti = np.random.normal(loc=70, scale=15, size=100)

print("### 100 Gözlemlü Veri Setinin İlk 5 Değeri ###")

print(veri_seti[:5])

ortalama = np.mean(veri_seti)

varyans = np.var(veri_seti, ddof=1)

standart_sapma = np.std(veri_seti, ddof=1)

ortalama_sapma = np.mean(np.abs(veri_seti - ortalama))

degisim_katsayisi = (standart_sapma / ortalama) * 100

print("\n" + "="*50)

print("### İSTATİSTİKSEL ÖLÇÜMLER (n=100) ###")

print("="*50)

print(f"1. Aritmetik Ortalama ($\bar{x}$): {ortalama:.2f}")

print(f"2. Varyans ($s^2$): {varyans:.2f}")

print(f"3. Standart Sapma ($s$): {standart_sapma:.2f}")

print(f"4. Ortalama Mutlak Sapma (OMS): {ortalama_sapma:.2f}")

print(f"5. Değişim Katsayısı (CV): {degisim_katsayisi:.2f}%")

print("="*50)
```

```

df_istatistik = pd.DataFrame({ 

    'Ölçüm': ['Ortalama', 'Varyans', 'Standart Sapma', 'Ortalama Mutlak Sapma', 'Değişim Katsayısı'], 

    'Değer': [ortalama, varyans, standart_sapma, ortalama_sapma, degisim_katsayisi] 

}) 

print("\n### Hesaplamaların Özeti (Pandas DataFrame) ###") 

print(df_istatistik.round(2))

```

## Kodun Açıklaması

### 1. Kütüphaneler

- **numpy**: Matris ve dizilerle çalışmak için temel kütüphanedir. Rastgele sayı üretme ve temel istatistiksel fonksiyonlar için kullanılır.
- **pandas**: Verileri tablo (DataFrame) formatında düzenli göstermek için kullanılmıştır.

### 2. Veri Seti Oluşturma

- `np.random.normal(loc=70, scale=15, size=100)`: 100 adet **rastgele sayı** üretir.
  - `loc=70`: Veri setinin **Ortalaması** 70 civarında olacak demektir.
  - `scale=15`: Veri setinin **Standart Sapması** 15 civarında olacak demektir.
  - `size=100`: **Gözlem sayısı** 100 olacaktır.

### 3. Hesaplamlar

**Varyans ve Standart Sapma:** np.var() ve np.std() fonksiyonlarında kullanılan **ddof=1** parametresi çok önemlidir. Bu,  $n-1$ 'e bölerek **örneklem** (sample) varyans ve standart sapmasını hesaplamınızı sağlar. İstatistik biliminde bu, daha doğru bir tahmin edici kabul edilir.

**Ortalama Mutlak Sapma (OMS):** NumPy'de doğrudan bir fonksiyonu olmadığı için formülle hesaplanmıştır:

- `veri_seti - ortalama`: Her gözlemin ortalamadan farkı bulunur.
- `np.abs(...)`: Farkların mutlak değeri alınır.
- `np.mean(...)`: Bu mutlak farkların ortalaması alınır.

**Değişim Katsayısı (CV):** Standart sapmanın ortalamaya oranlanıp 100 ile çarpılmasıyla  $(s/x^*100)$  elde edilir.

Bu kodu çalıştırıldığınızda, 100 gözlemli veri setiniz için gerekli olan tüm yayılım ölçümlerini yanında elde edeceksiniz.

```
### 100 Gözlemli Veri Setinin İlk 5 Değeri ###
[77.4507123 67.92603548 79.71532807 92.84544785 66.48769938]

=====
### İSTATİSTİKSEL ÖLÇÜMLER (n=100) ###
=====
1. Aritmetik Ortalama (ar{x}): 68.44
2. Varyans ($s^2$): 185.57
3. Standart Sapma ($s$): 13.62
4. Ortalama Mutlak Sapma (OMS): 10.77
5. Değişim Katsayısı (CV): 19.90%
=====

### Hesaplamların Özeti (Pandas DataFrame) ###
      Ölçüm    Değer
0      Ortalama  68.44
1      Varyans  185.57
2  Standart Sapma  13.62
3  Ortalama Mutlak Sapma  10.77
4      Değişim Katsayısı  19.90
```

Windows

Sunulan 100 gözlemli veri setinin analiz sonuçları, incelenen değişkenin merkezi eğilim ve yayılım düzeyleri hakkında önemli çıkarımlar sağlamaktadır. Veri setinin Aritmetik Ortalaması (68.44), gözlemlerin tipik değerinin bu puan etrafında merkezlendiğini göstermektedir. Dağılımın değişkenliğini mutlak birimlerle ölçen Standart Sapma (13.62) değeri, bireysel gözlemlerin ortalamadan yaklaşık 13.62 birimlik bir sapma gösterdiğine işaret eder. Bu veri setinin Varyansı (185.57), yayılımın karesel ölçüsünü sunarken; Ortalama Mutlak Sapma (10.77), ortalamadan sapmaların mutlak değerlerinin

ortalaması olup, Standart Sapmaya yakınlığı, aykırı değerlerin etkisinin sınırlı olduğu nispeten simetrik bir dağılım olasılığını desteklemektedir. Son olarak, (19.90%) olarak hesaplanan Değişim Katsayısı, standart sapmanın ortalamaya oranını gösteren ve birimsiz olan göreli bir yayılım ölçüsü olarak, bu veri setinin değişkenlik düzeyinin yüzde yirmiye yakın olduğunu ve dolayısıyla farklı ortalamalara sahip diğer veri setleriyle karşılaştırılabilir orta düzeyde bir değişkenlik sergilediğini ortaya koymaktadır.