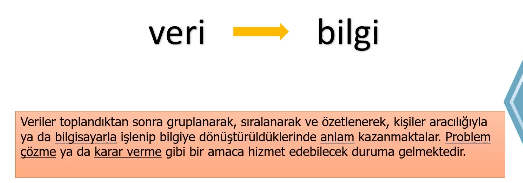
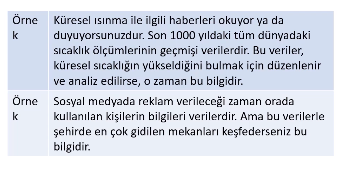
**2\_VERİ OKURYAZARLIĞI**

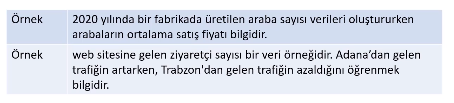
**2.1\_Veri Nedir?**

Akıllı cihazlar aracılığıyla yaptığımız işlemlerin bilgisayar programları tarafından algılanabilmesini sağlamak amacıyla formüle edilmiş haline **veri** denir.



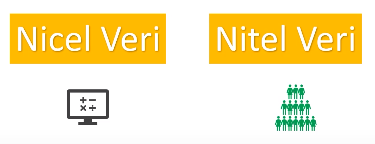


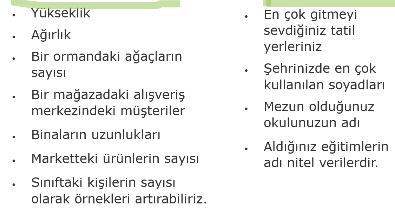


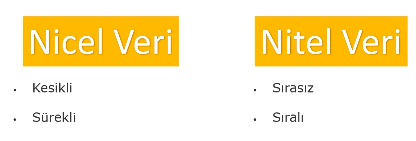




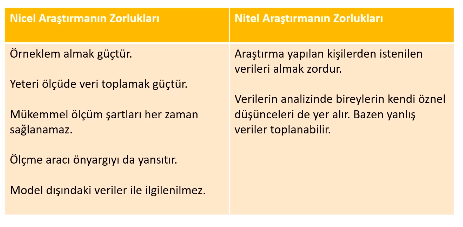
**2.2\_Veri Türü Nedir?**

****

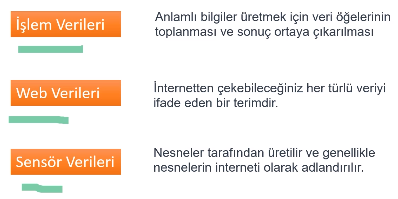
****

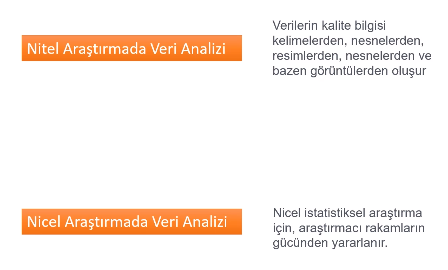
****

****

****

**2.3\_Veri Toplama Yöntemleri Nedir?**

****

****

**2.4\_Veri Madenciliği**

1700’lü yıllarda regresyon analizinin keşfiyle ortaya çıkmıştır.



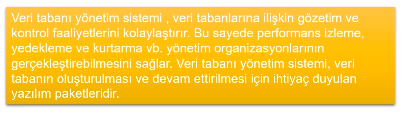




**2.5\_Veri Tabanı Nedir?**

****

**2.6\_Veri Tabanı Yönetim Sistemi**

****

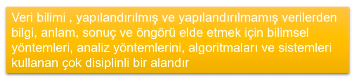
**2.7\_En Çok Kullanılan Veri Tabanı Sistemleri**

****

**2.8\_Veri Merkezleri**

****

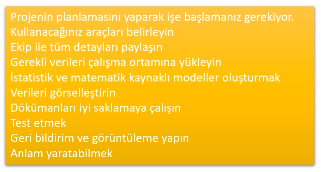
**2.9\_Veri Bilimi Nedir?**

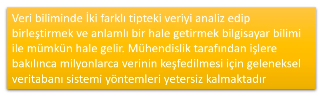
****

****

****

****

****

****

**2.10\_Veri Bilimi Tarihçesi**

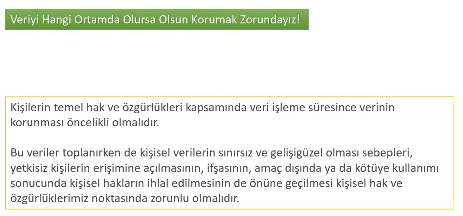
****

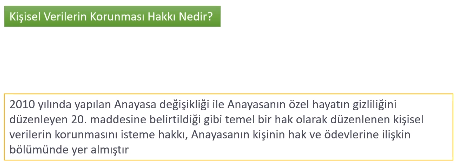
****

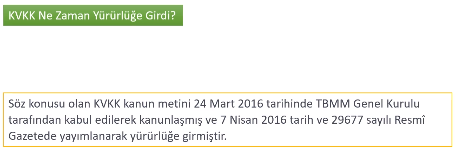
**2.11\_Veri Bilimi ve Diğer Bilimlerle İlişkisi**

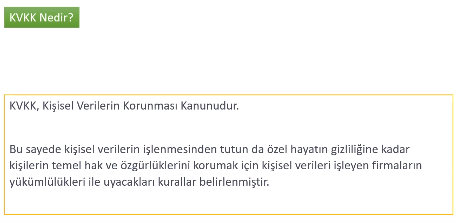
****

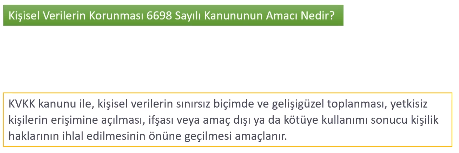
**2.12\_KVKK 1. Bölüm**

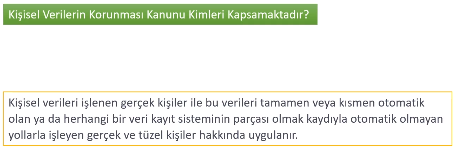
****

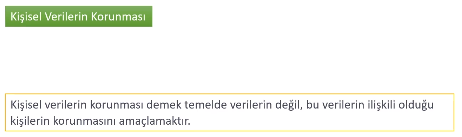
****

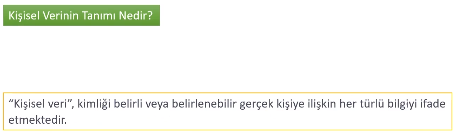
****

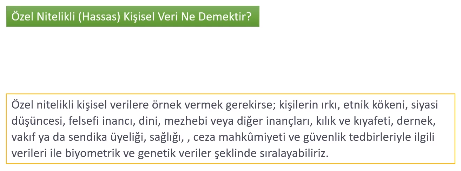
****

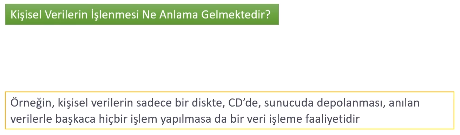
****

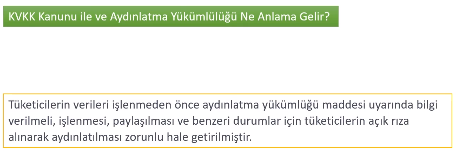
****

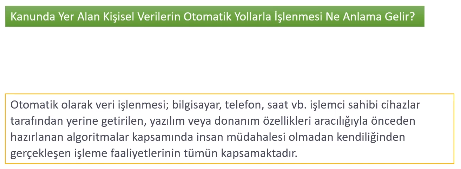
****

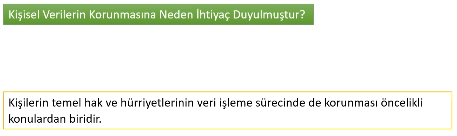
****

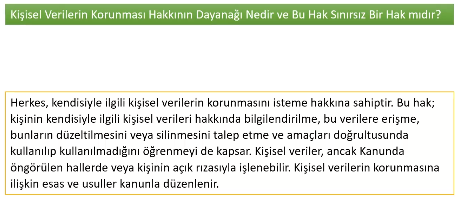
****

****

****

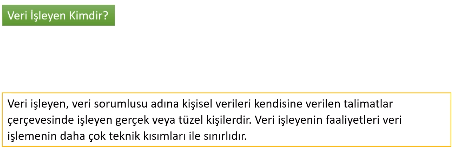
****

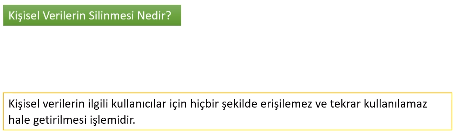
****

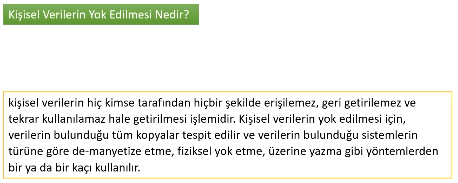
****

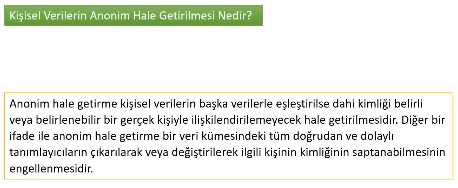
**2.13\_KVKK 2. Bölüm**

****

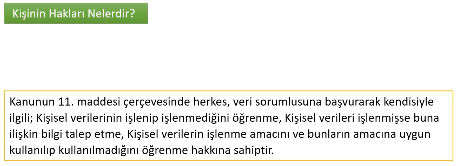
****

****

****

****

****

****

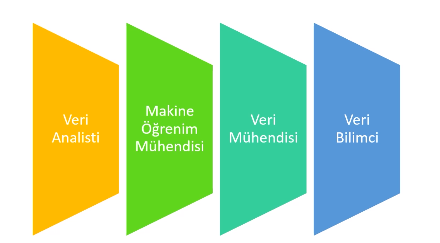
**2.14\_Veri Bilimi Araçları Nelerdir?**

****

**2.15\_Veri Biliminden Hangi Alanlarda Yararlanılıyor?**

****

**2.16\_Veri Bilimi Kariyer Alanları Nelerdir?**

****

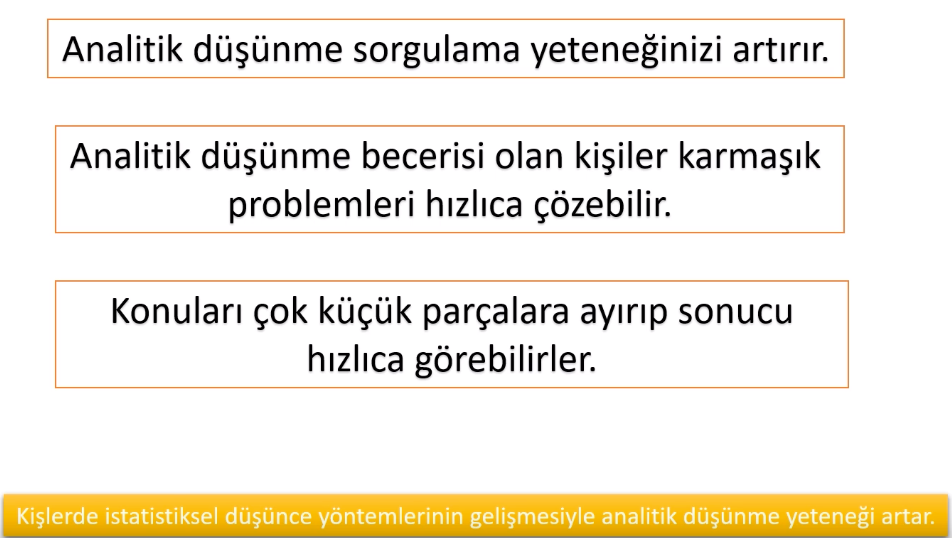
**Not:** Bunlara ek olarak “Veri Görselleştirme Uzmanı, Veri Tabanı Uzmanı” alanları da mevcuttur ve sadece bunlarla sınırlı değildir.



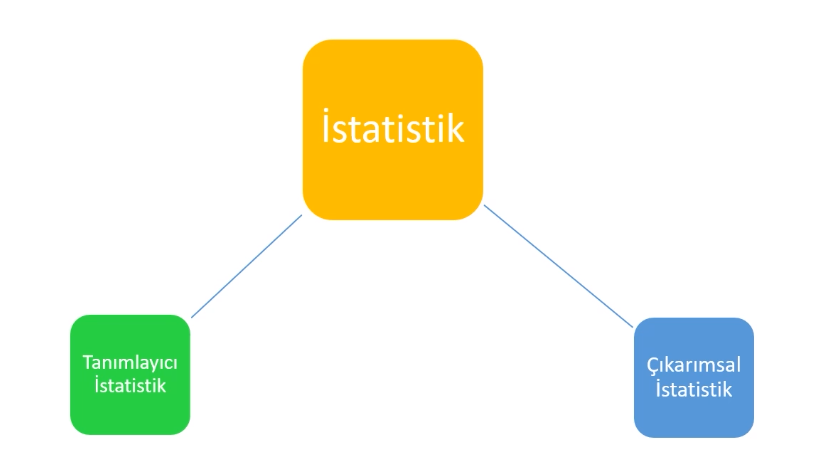
**3\_TEMEL İSTATİSTİK**

**3.1\_İstatistiksel Düşünce**

****

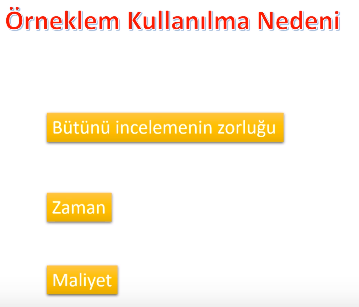


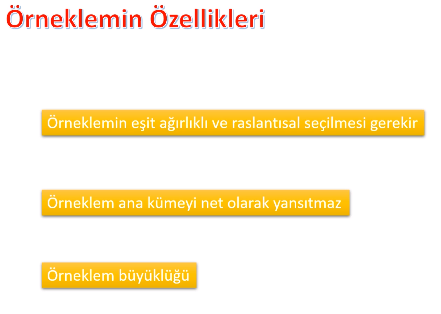




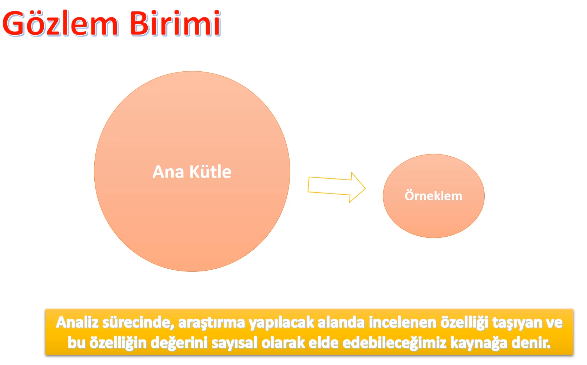
**3.2\_Örneklem**

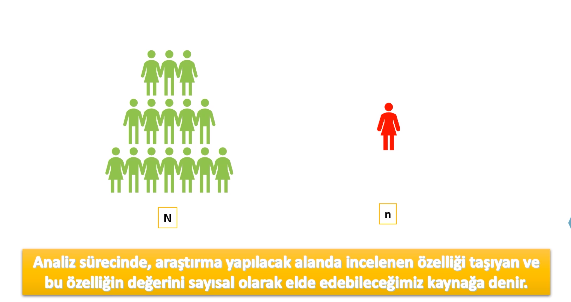




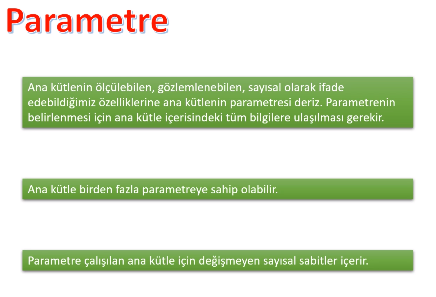


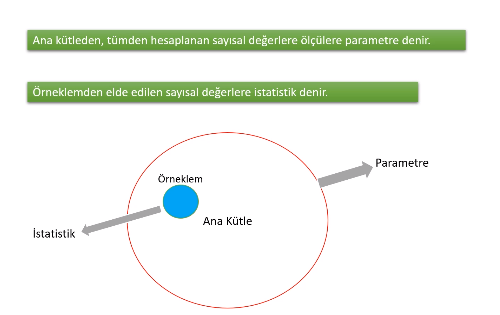
**3.3\_Gözlem Birimi**

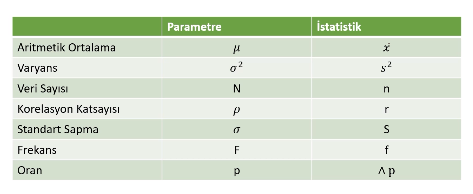




**3.4\_Parametre**

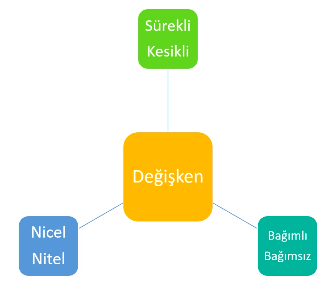
****

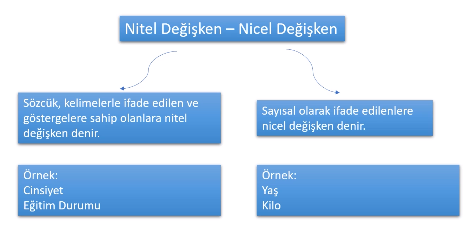
****

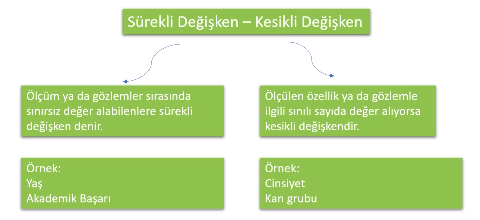
****

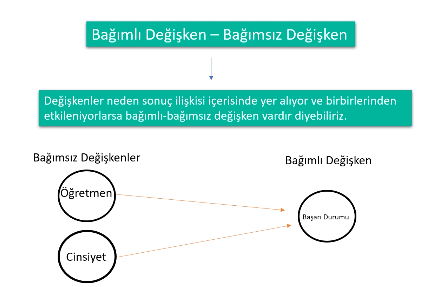
**3.5\_Değişkenler**

****

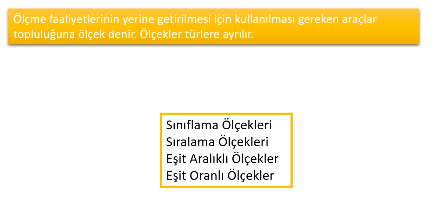
****

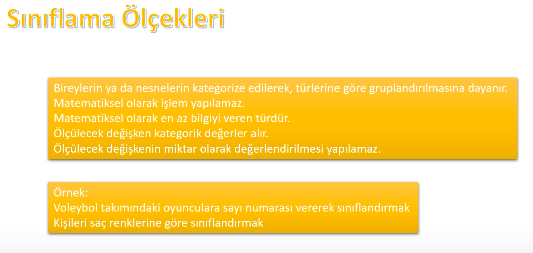
****

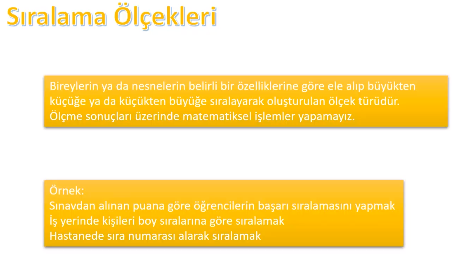
****

****

**3.6\_Ölçek Türleri**

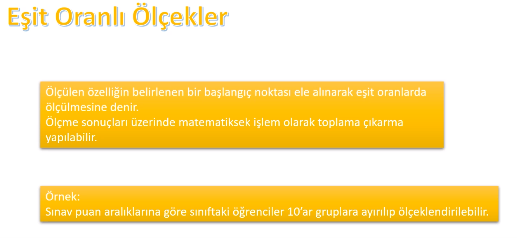
****

****

****

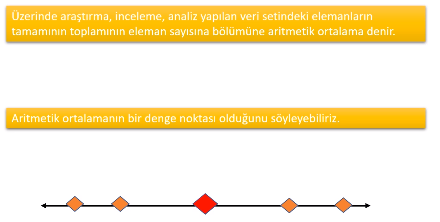
****

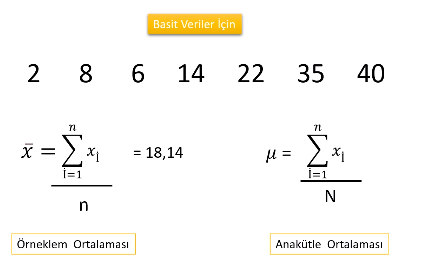
**Not: 0’dan küçük değerlerden de oluşabilecek ölçek türüdür.**

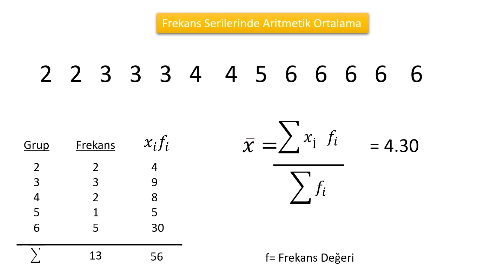
****

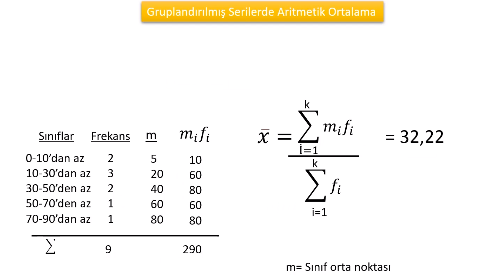
**Not: Başlangıç noktası 0 olan ölçek türüdür.**

**3.7\_Aritmetik Ortalama**

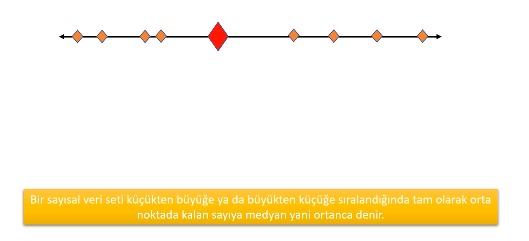
****

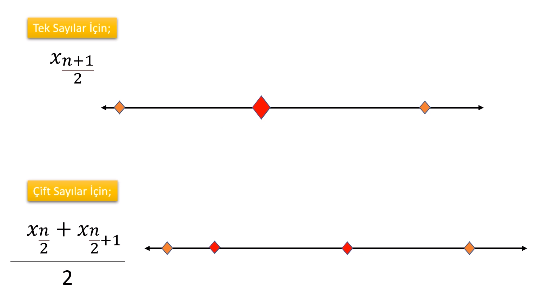
****

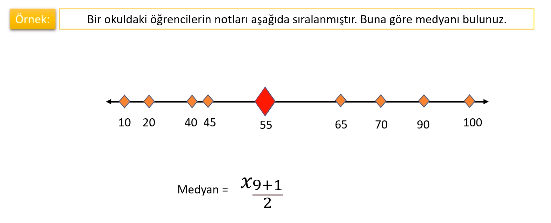
****

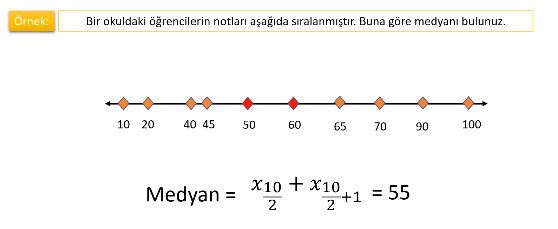
****

**3.8\_Medyan**

****

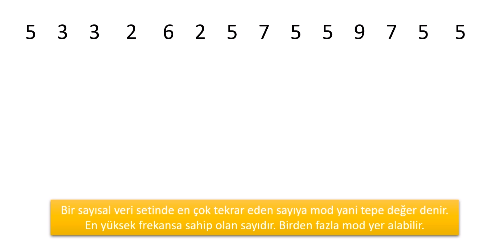
****

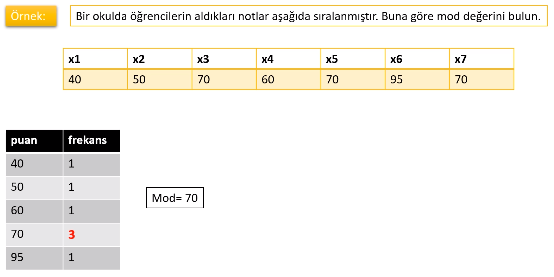
****

****

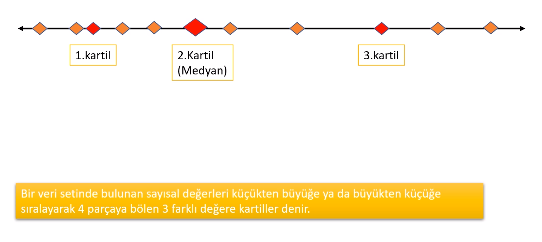
**Not: Buradaki formül gerçek değeri değil, terim sayısını (numarasını) verir.**

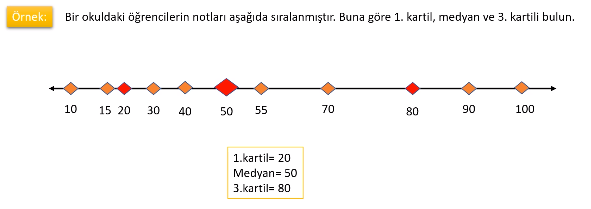
**3.9\_Mod**

****

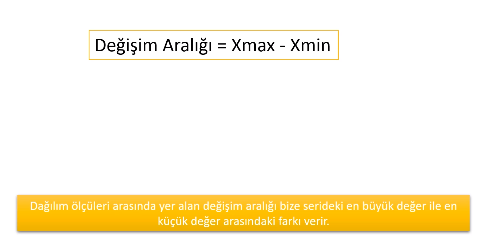
****

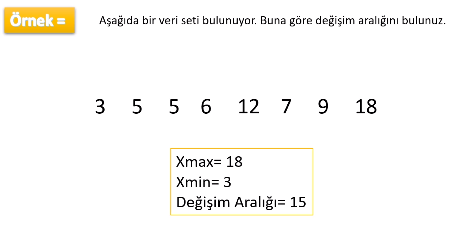
**3.10\_Kartiller**

****

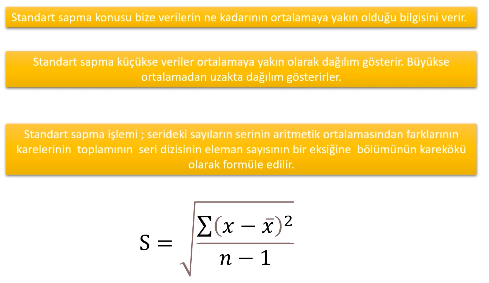
****

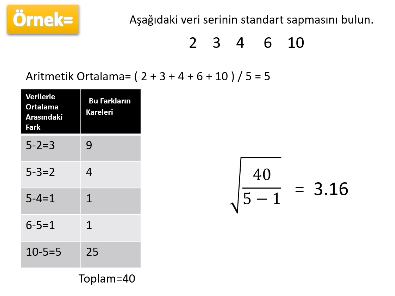
**3.11\_Değişim Aralığı**

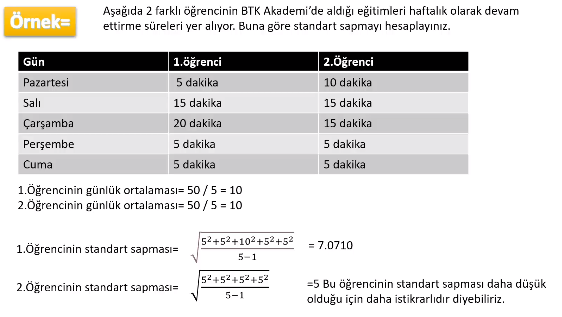
****

****

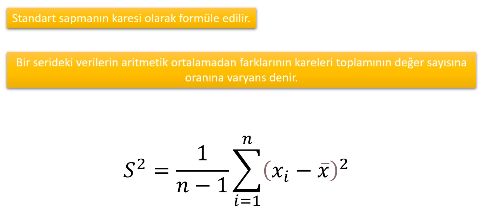
**3.12\_Standart Sapma**

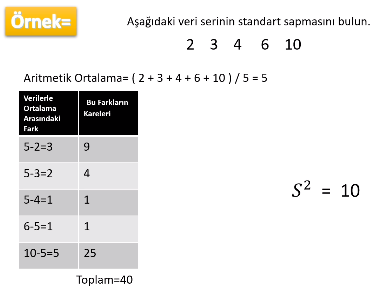
****

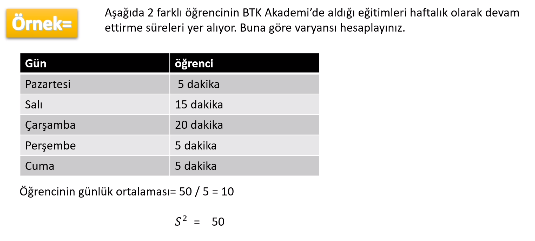
****

****

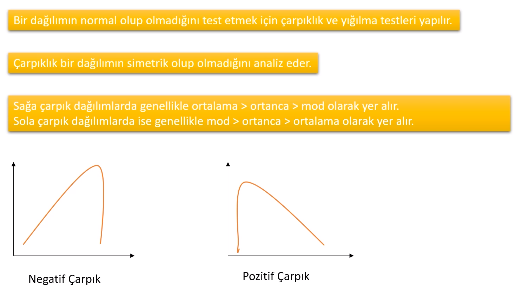
**3.13\_Varyans**

****

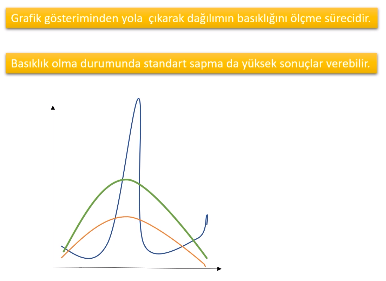
****

****

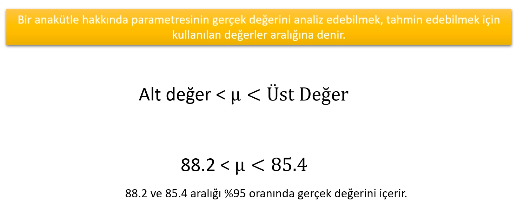
**3.14\_Çarpıklık**

****

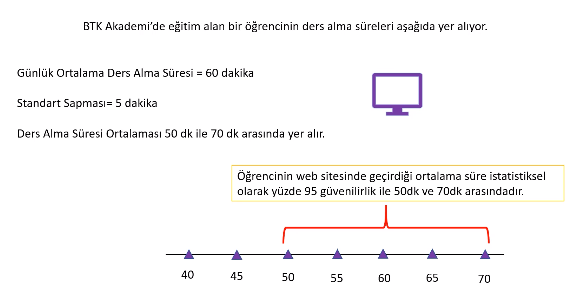
**3.15\_Basıklık**

****

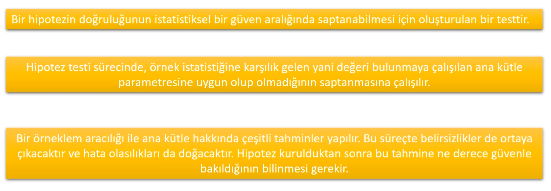
**3.16\_Güven Aralıkları**

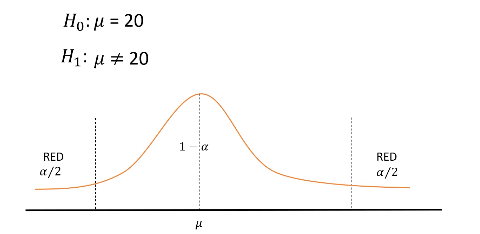
****

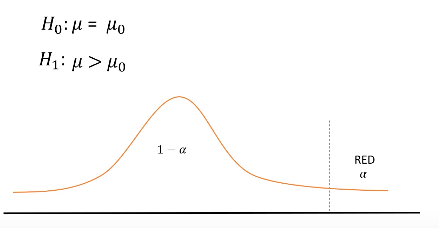
**Not: Güven aralıkları “µ” ile ilişkilendirilir.**

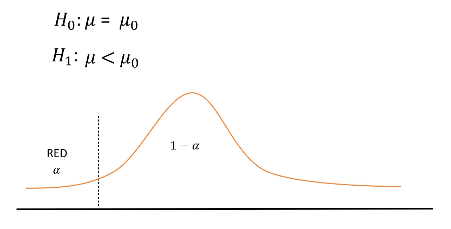
****

**3.17\_Hipotez Testleri**

****

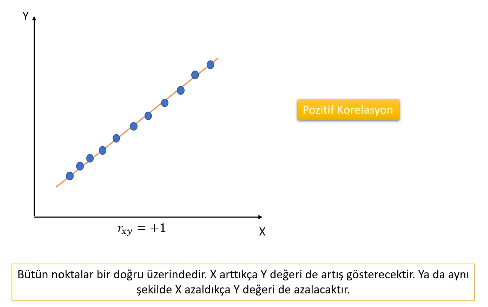
****

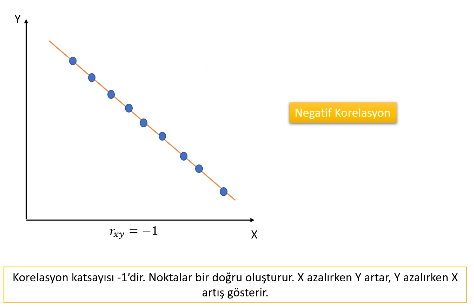
****

****

**3.18\_Korelasyon**

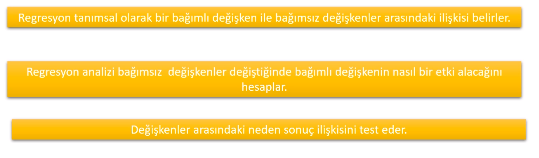
****

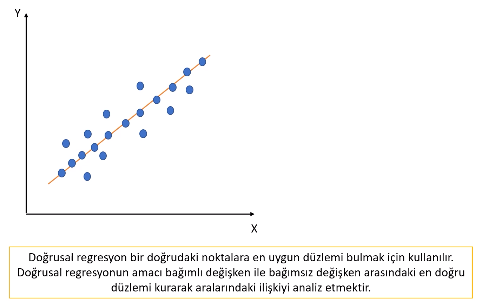
****

****

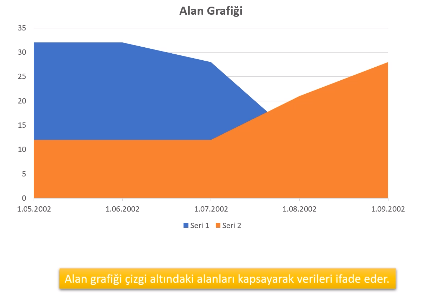
****

**3.19\_Regresyon**

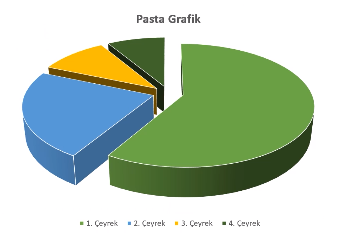
****

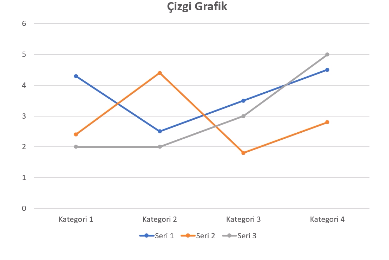
****

**3.20\_Grafik Verilerinin Okunması**

****

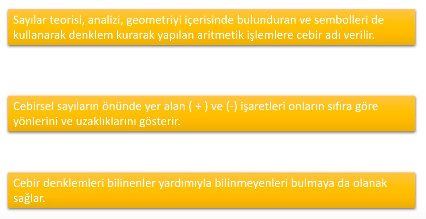
****

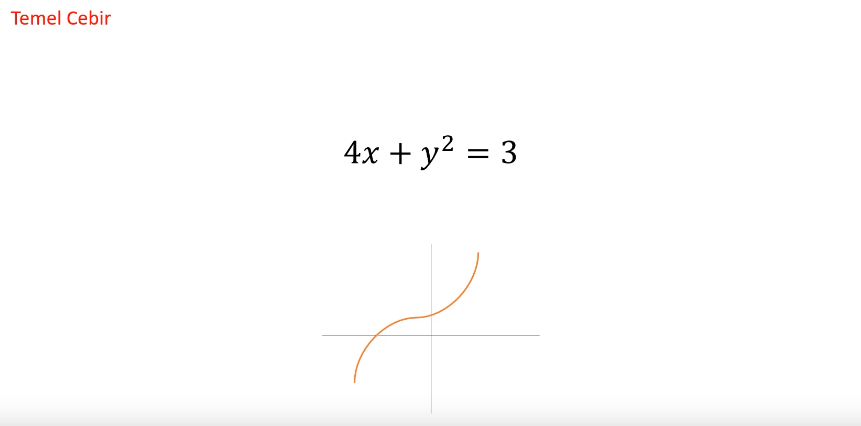
****

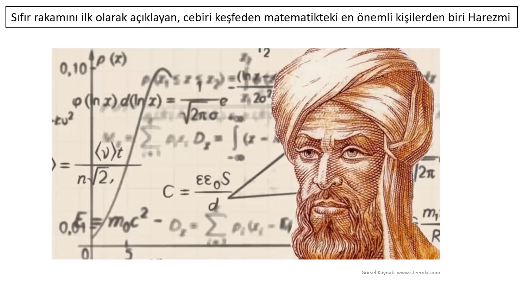
****

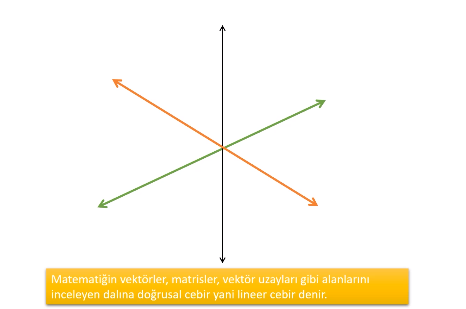
**4\_LİNEER CEBİR**

**4.1\_Lineer Cebir Nedir?**

****

****

****

****

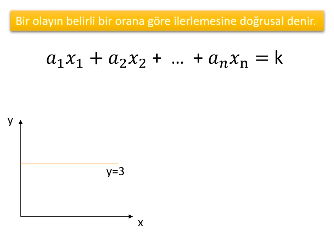
****

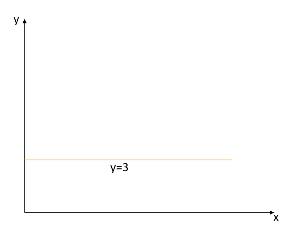
**4.2\_Lineer Cebir Nerelerde Kullanılır?**

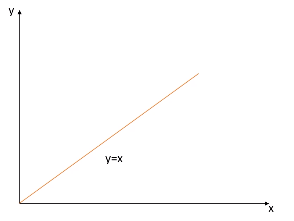
****

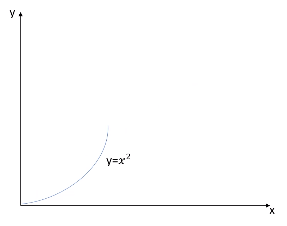
****

**4.3\_Lineer Denklem Özellikleri**

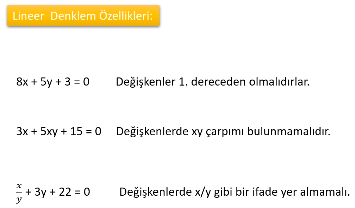
****

****

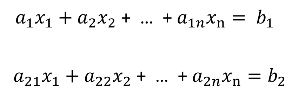
****

****

**Bu grafik doğrusallık özelliği taşımamaktadır.**

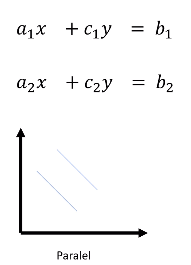
****

**4.4\_Lineer Denklem Sistemi**

****

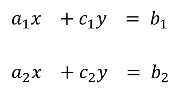
Lineer cebir. Her denklemde farklı bir bilinmeyende karşımıza çıkmaktadır. Örneğin havuz ve yaş problemlerinde karşımıza çıkan denklem sistemleriydi. Aslında Lineer Cebir çözmesi zor olan problemlerin matrisler aracılığıyla kolay bir şekilde çözülebiliceğini bize göstermektedir. Daha da karmaşık olan yapay zeka,makine öğrenmesi gibi çok farklı denklemler içeren bir bilgisayar programlamasındaki denklemleri çok hızlı bir şekilde çözebiliriz.

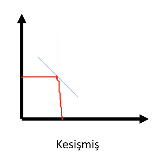
**Doğrusal denklemlerde parelellik:**

****

**Burada x’in katsayıları(a1 ve a2) eşittir. Yani bu katsayıların eşit olması durumunda paralellik özelliği gösterir.**

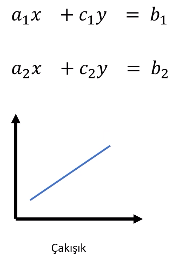
**Kesişme özelliği:**

****

****

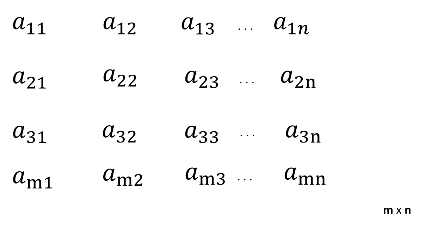
**Burada x ve y’nin ortak bir noktada birleşmesiyle kesişme özelliği sağlanır. Böyle durumlarda iki denklemi birbirine eşitleyerek işlem yapmak gerekiyor.**

**Çakışık özelliği:**

****

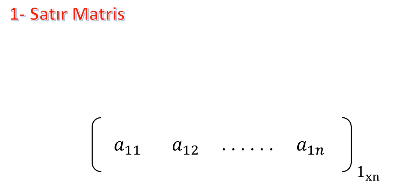
**Böyle durumlarda ise işlemimize şeklinde devam etmemiz gerekiyor. Yani burada x’in, y’nin ve b’nin katsayılarının eşit olduğunu varsayıyoruz.**

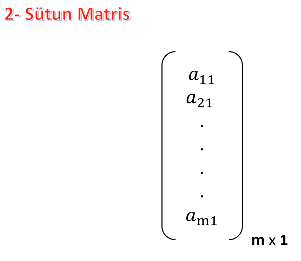
**4.5\_Matris Çeşitleri 1**

****

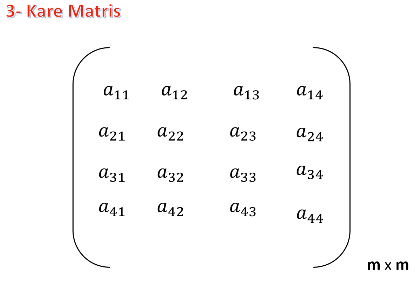
**Lineer denklem sistemindeki katsayıları barındıran ve bununla işlemler yaptığımız ifadeye matris denir. (m = satır, n = sütun)**

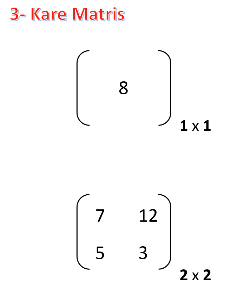
**10 tane matris türü vardır:**

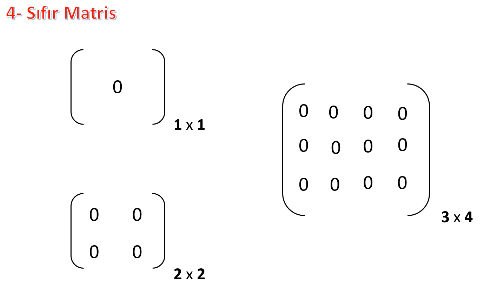
****

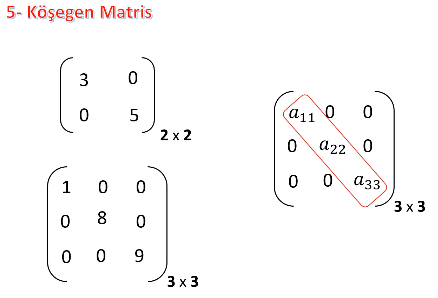
****

**4.6\_Matris Çeşitleri 2**

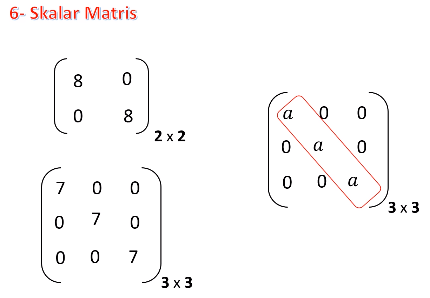
****

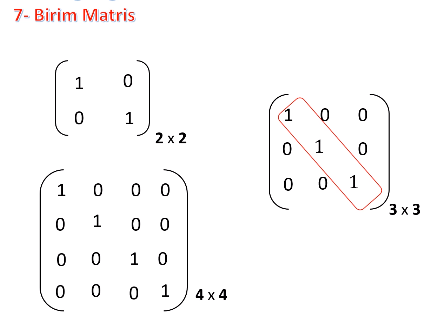
****

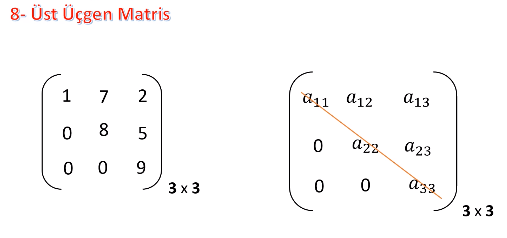
****

****

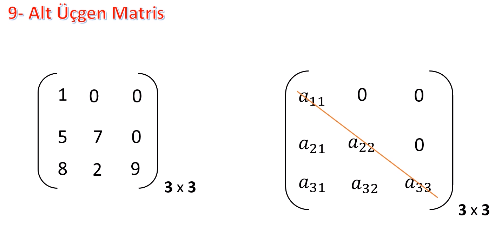
**Köşegenler dışındaki tüm sayılar sıfırdır.**

****

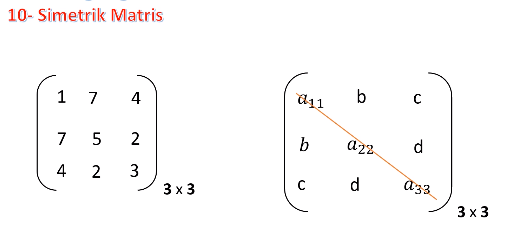
****

****

**Üst üçgen şeklinde çizilebilen matristir. Onun dışındaki sayılar sıfır olmalıdır.**

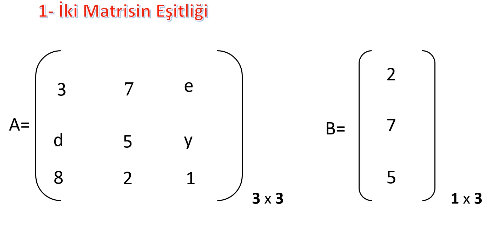
****

**Alt üçgen şeklinde çizilebilen matristir. Onun dışındaki sayılar sıfır olmalıdır.**

****

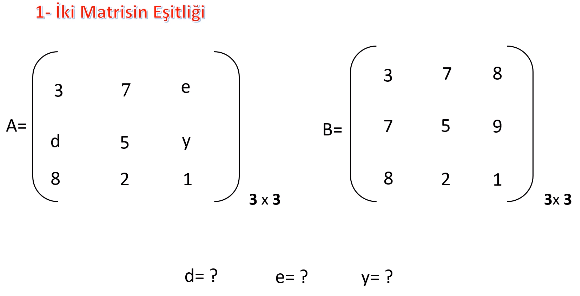
**4.7\_Matrisle İlgili Terimler**

****

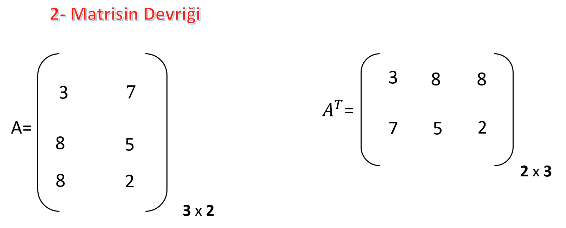
****

**Bu matrislerin eşit olmadığını söyleriz. Çünkü satır-sütunları ve içerisindeki sayılar farklıdır.**

**Not: B matrisi yanlış yazılmış, 3x1 olmalı.**

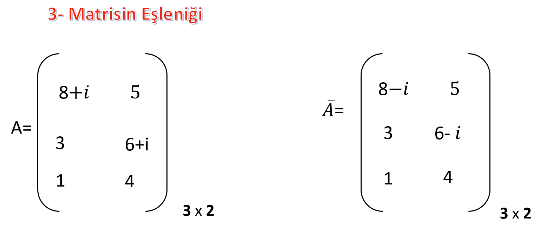
****

**Burada ise görüldüğü gibi satır-sütunlar ve sayılar eşit olduğu için A ve B matrisleri eşittir diyebiliriz. Bilinmeyenler yerine B matrisinin aynısı yazılabilir.**

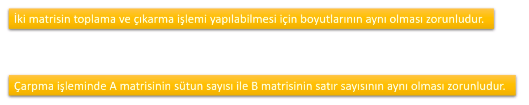
****

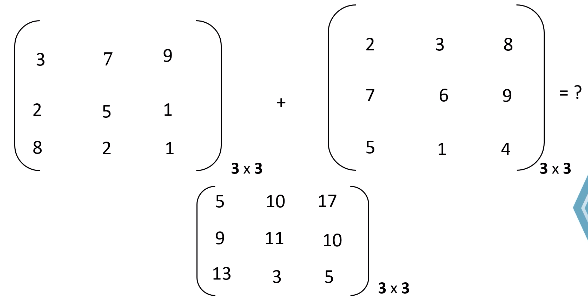
**Matrisin transpozu da denir. Satır ve sütunun yer değiştirilmesiyle bu matris elde edilir.**

**( … haline gelir.)**

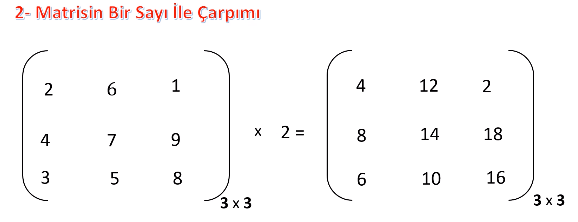
****

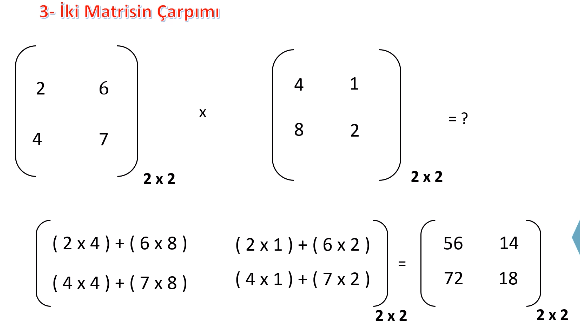
**4.8\_Matriste İşlemler 1**

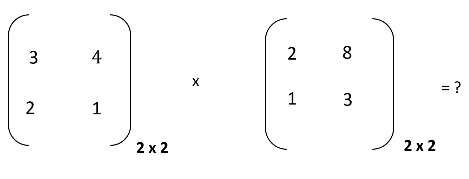
****

****

**4.9\_Matriste İşlemler 2**

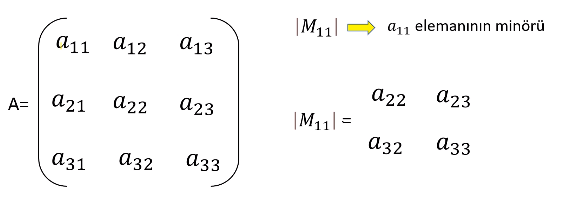
****

****

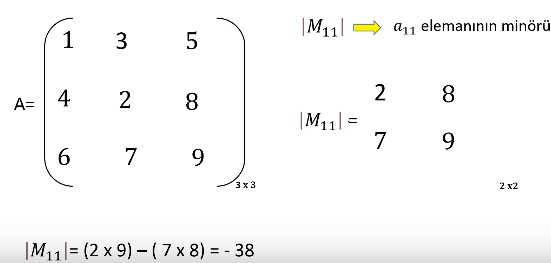
****

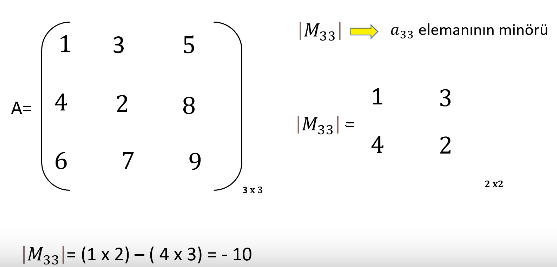
**4.10\_Matrisin Eş Çarpanı**

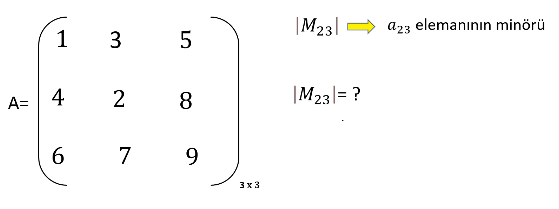
**Minör:**

****

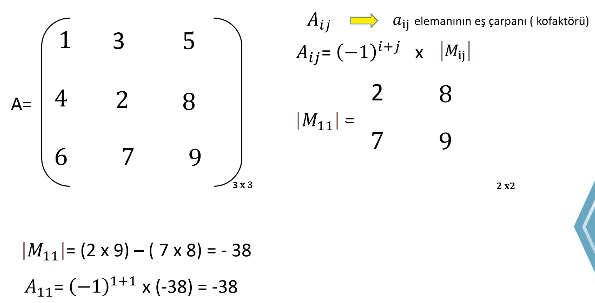
Determinant hesabında kullanılacaktır.

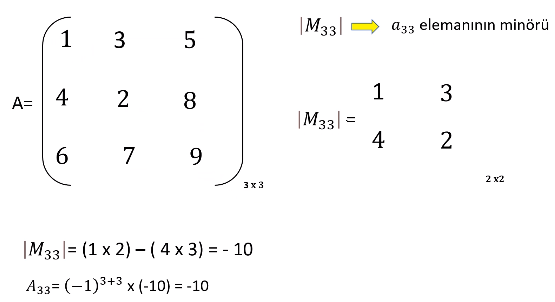


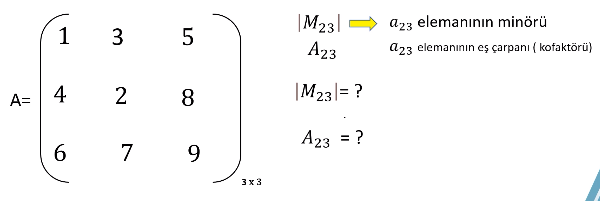




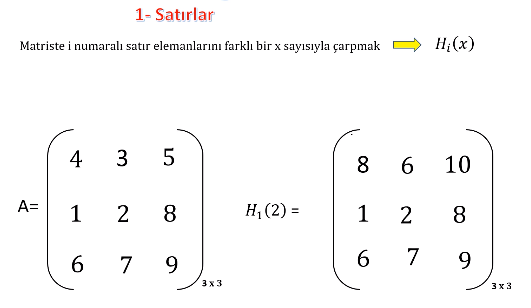
**Kofaktör:**

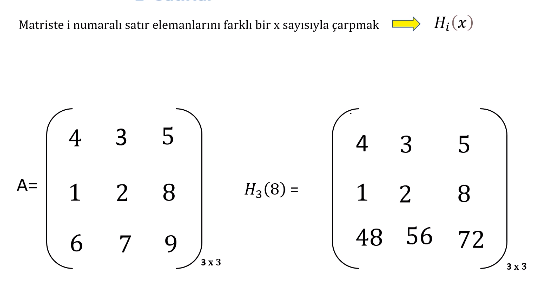




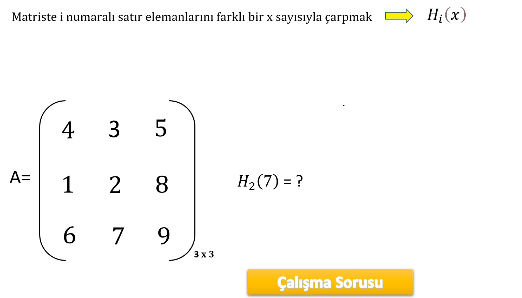


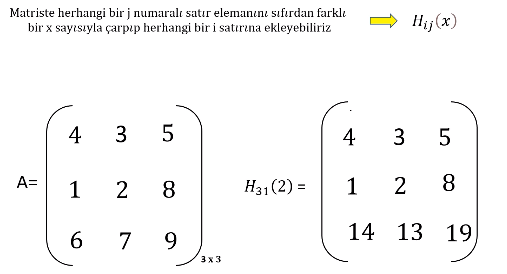
**4.11\_Matrislerde İndirgeme 1**

****

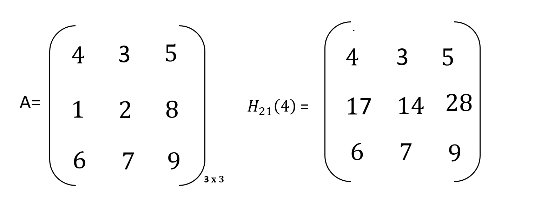
****

Matrisin satırlarının yerini değiştirmek istiyorsak yazım tarzı şeklinde olup bu örneğe göre 1. ve 2. Satır yer değiştirilerek yazılmalıdır.





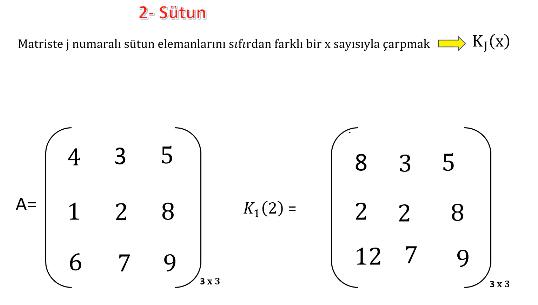
Buradaki işlem 1. satırdaki elemanları 2 ile çarpıp çıkan sonuca 3. satırdaki elemanlar eklenir ve 3. satıra yazılır.

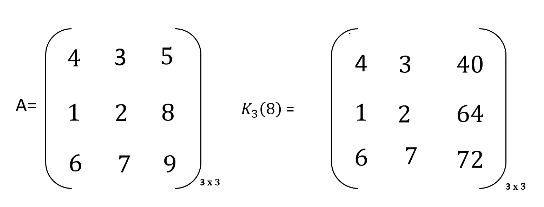


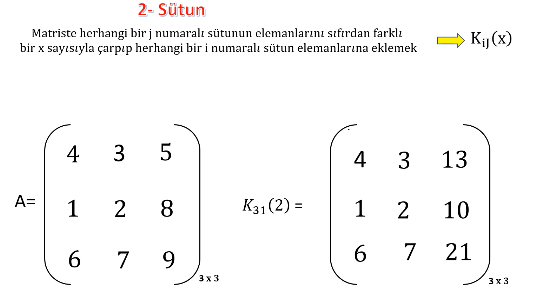
Buradaki işlem 1. satırdaki elemanları 4 ile çarpıp çıkan sonuca 2. satırdaki elemanlar eklenir ve 2. satıra yazılır. Yani H’den hemen sonra gelen sayı(i) neyse o satıra işlem yapılacaktır.



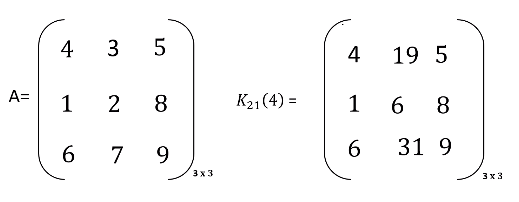
**4.12\_ Matrislerde İndirgeme 2**

****

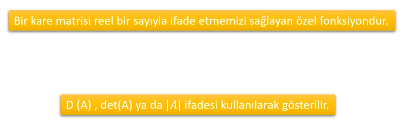
****

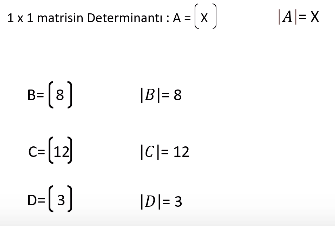
****

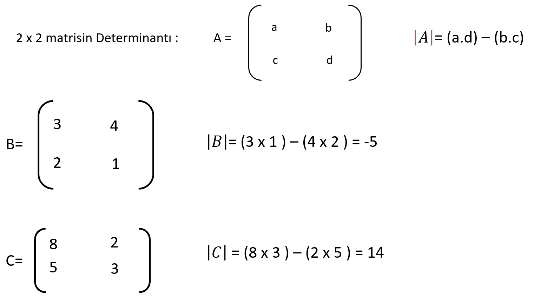
Buradaki işlem 1. sütundaki elemanları 2 ile çarpıp çıkan sonuca 3. satırdaki elemanlar eklenir ve 3. sütuna yazılır.



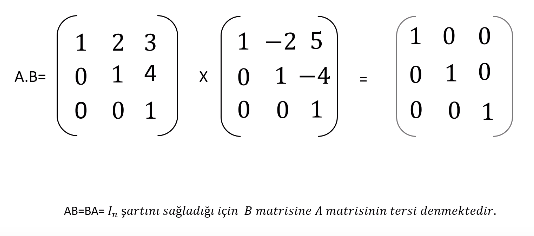
**4.13\_Determinant**

****

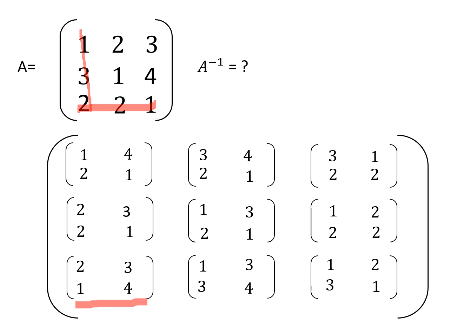
****

****

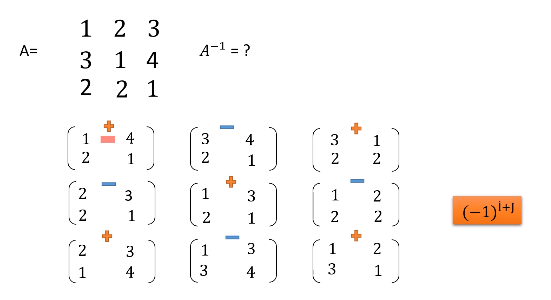
**4.14\_Matrisin Tersi**

****

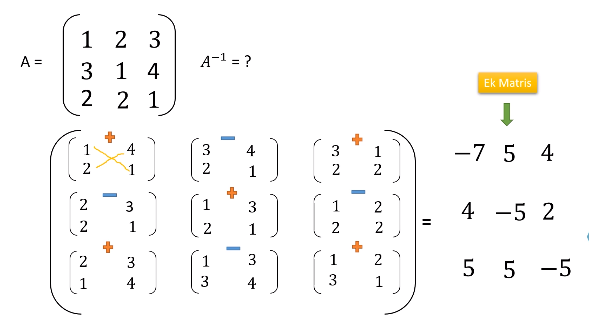
**Adım 1:**

****

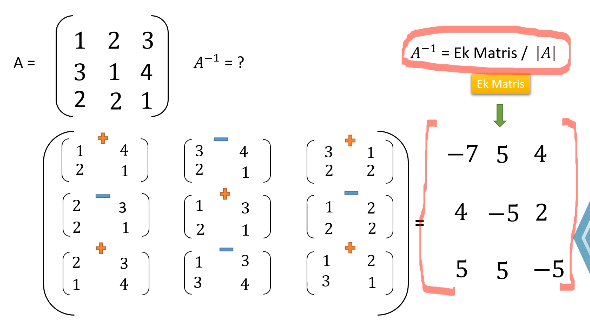
**Adım 2:**

****

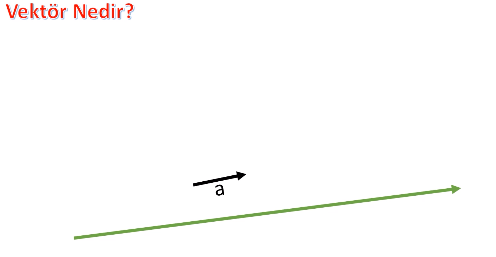
**Adım 3:**

****

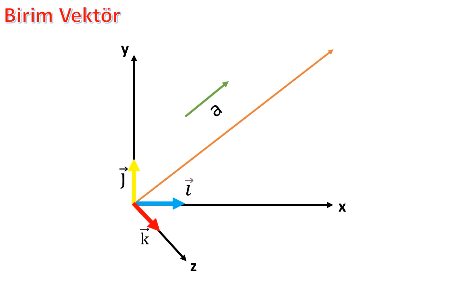
**Adım 4:**

****

**4.15\_Vektörler Giriş**

****

Sayıları, nesneleri, işlemleri sürdürürken buradaki işlemlerin sayısal değerinin ötesinde yönlerini de ele almamız gerekiyor. Burada vektörler bunu sağlıyorlar. Algoritma, bilgisayar programlama, mühendislik, haritacılık gibi alanlarda, akıllı cihazlar gibi uzaktan iletişim kurmamızı sağlayan yani sinyal yoluyla çalışan araçlarda da kullanılır. Şekilde a olarak ifade edilen değerin üstündeki ok işareti, a değerinin bir vektörel büyüklüğe sahip olduğunu göstermektedir.

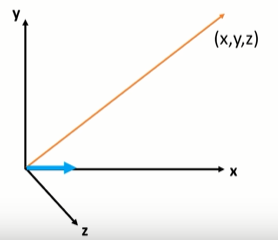


Vektörlerin bulundukları yerlere göre nasıl bir konumda olduklarını belirtebiliyoruz, buna göre referans alabiliyoruz. Örneğin bir tahtaya x ve y eksenini çizerek a vektörünü koysaydık 2 boyutlu bir alanda yer aldığını söyleyebilirdik. Fakat bir z ekseni de çizildiğinde vektörün 3 boyutlu bir alanda da yer aldığını söyleyebiliriz(adlandırma yapabiliriz). Birim vektörler, x ekseninde vektörümüzün nasıl hareket aldığı ile alakalı ya da y ekseninde bulundurduğu hareketler veya z ekseninde bulundurduğu hareketler sonucunda farklı değerler alabilir. x ekseninde i olarak, y ekseninde j olarak, z ekseninde k olarak vektörün yönelimi belirleyebiliriz ve vektörün nasıl bir büyüklüğe/boyuta ulaştıklarını hesaplayabiliriz.

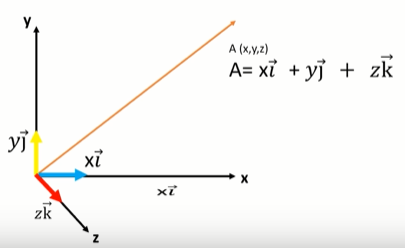
**4.16\_Yer Vektörü**

****

Başlangıç noktası orijinde olan vektörlerdir. Yer vektörleri birim vektörlerin birleşimidir.



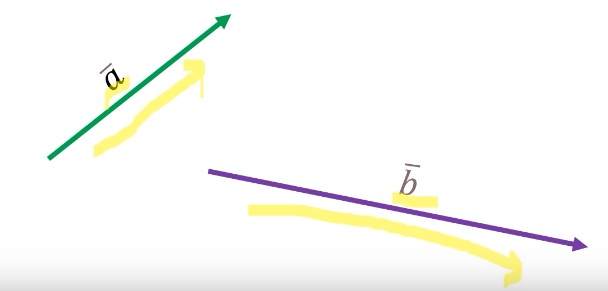
Uzayda koordinatları yani konumu bilinen iki nokta arasındaki uzaklık ve yönünü bulabilmek için yer vektörü kullanılır. Eğer sadece bir noktanın koordinatları biliniyorsa konum vektörünün ordinatlarını (0,0,0) olarak çizilen bir vektörle ifade edebiliriz. Şekilde görüldüğü üzere orijinden geçen bir vektör bulunmaktadır.



Grafikleri vektörler nasıl oluşuyor dersek burada i vektörü x kadar ilerlemiş demektir. Aynı şekilde diğerleri de (j ve k vektörleri) y ve z kadar ilerlemiştir. A vektörünün oluşması ise şekilde formülle bulunmaktadır.

**4.17\_Vektörlerde Toplama Çıkarma**

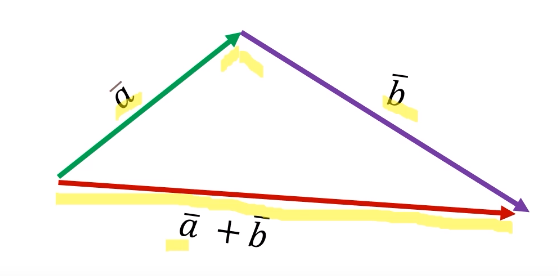
**Toplama İşlemi:**



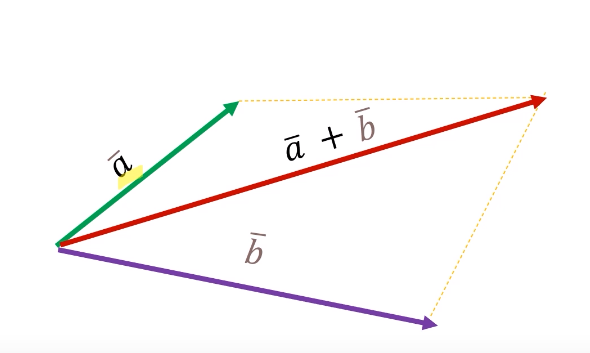
Bu iki vektörün farklı yöne gittiğini ve boyutunun farklı olduğunu görebiliyoruz.

Vektörlerde 2 farklı şekilde toplama işlemi yapabiliriz:

1. **Yöntem:**



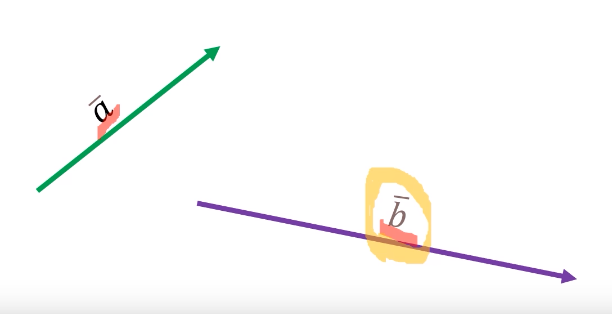
1. **Yöntem:**



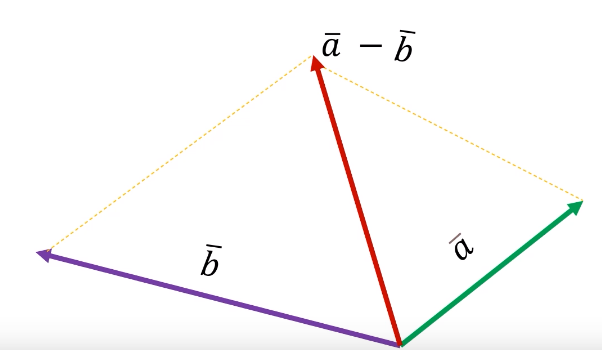
**Not:** Toplama işleminde vektörlerin yönleri değiştirilmemelidir. Çünkü her iki vektörün işareti de artıdır.

**Çıkarma İşlemi:**

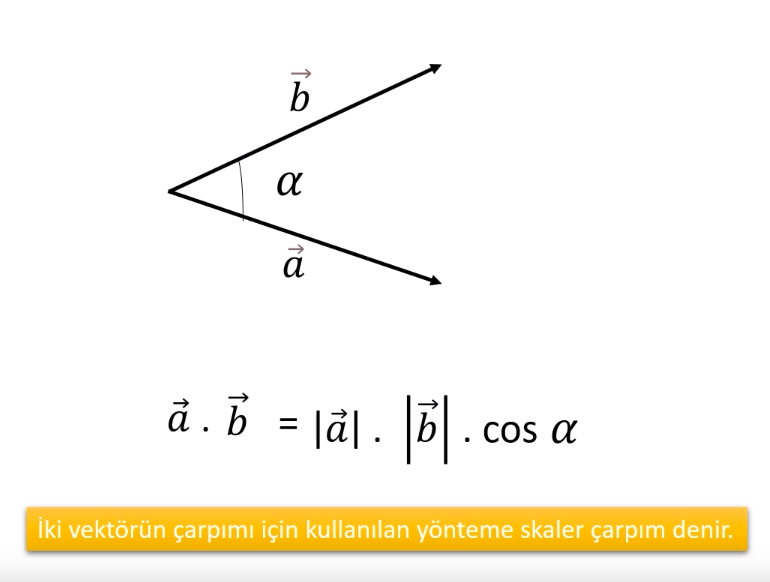
Negatif değer alacak olan vektörün yönünü değiştirerek çıkarma işlemi yapabiliriz. Vektörün ilk hali aşağıdaki gibiydi:



Çıkarma işlemi yaparken b vektörünün yönünü aşağıdaki gibi değiştirerek çıkarma işlemini yapabiliriz. Çünkü b’nin işareti eksi olduğu için yönü değişecektir.

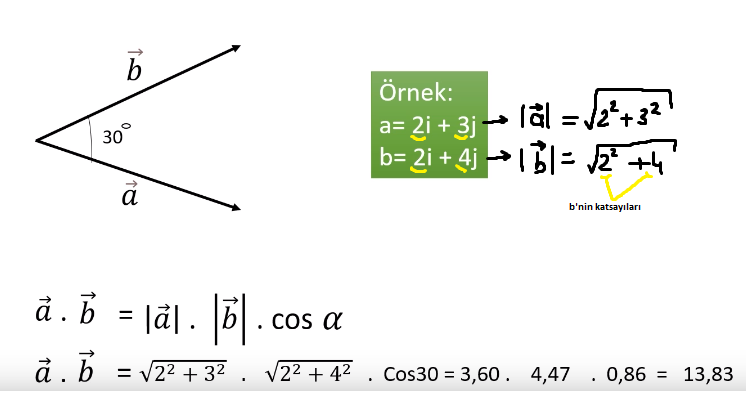


**4.18\_Skaler Çarpım**

****

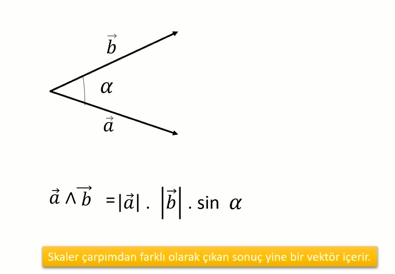
Burada amaç skaler bir sayıya ulaşmaktır.

3 boyutlu problemlerde çözüme ulaşmak için vektör yöntemleri uygulanmaktadır.

****

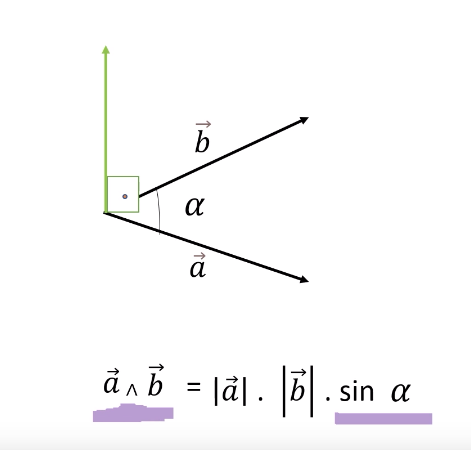
Kısacası a’nın ve b’nin katsayılarını karekök içinde karesini alıp toplarsak ve cos(a) ile çarparsak sonuca ulaşabiliriz.

**4.19\_Vektörel Çarpım**

****

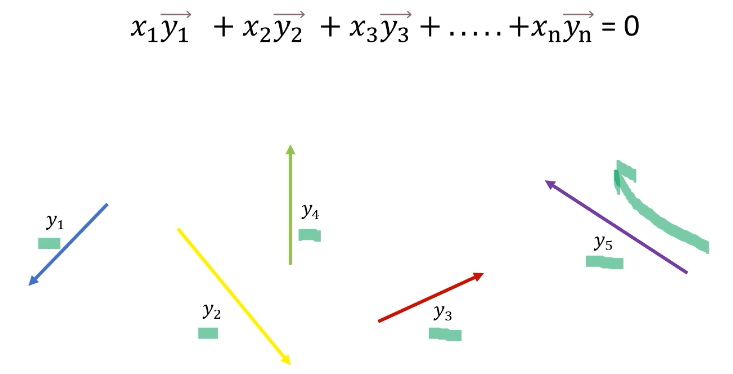
Skaler çarpımda sonuç skaler olarak çıkar ancak burada vektörel çarpımda sonuç vektör olarak çıkacaktır.

Aşağıdaki gibi ilk olarak 90 derece dik indirmemiz lazım:

****

**4.20\_Lineer Bağımlı**

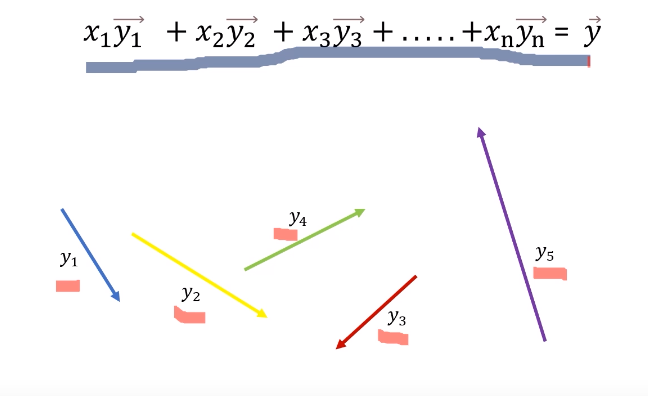
Eğer vektörlerimiz katsayılarımız ile toplandığında eğer sıfıra eşitse “Lineer Bağımlı” deriz.

****

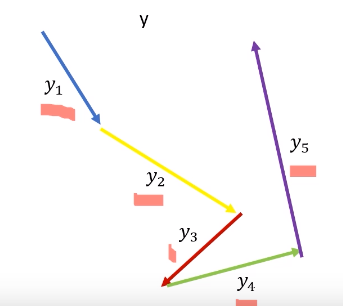
Aşağıda gördüğümüz şekilde uç uca eklenerek ve yönünü değiştirmeden kapalı bir çerçeve oluşuyorsa biz buna “Lineer Bağımlı” diyebiliriz.

****

Bağımsızlık konusunu ele alalım:

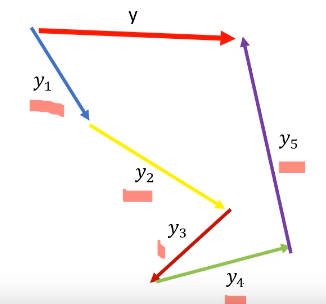
****

Aşağıdaki resimde de görüldüğü gibi kapalı alan oluşturmadı:

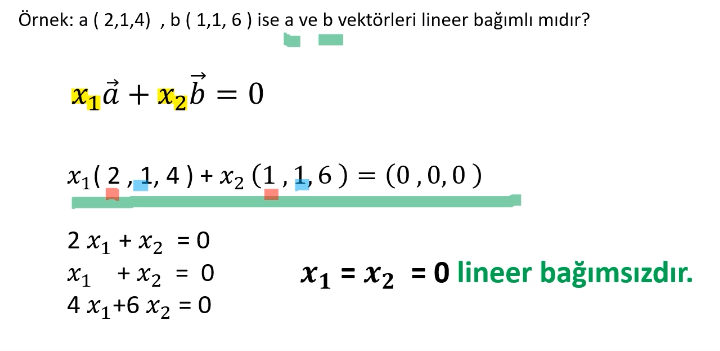
****

y vektörünü kullanarak kapalı alan oluşturabiliriz bu sebeple de vektörler toplamı y’ye eşit olmuş olur.

(Burada y vektörünün ters yönde olduğunu gözden kaçırmamak gerekir. Yani y5 vektörünün devamı niteliğinde değildir.)

****

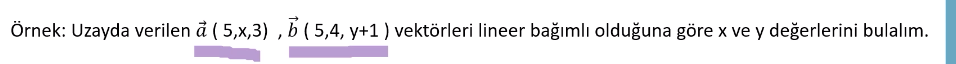
Burada “Lineer Bağımsızlık” söz konusudur diyebiliriz.

****

**Yanlış yazılmış!**

Lineer bağımlıdır olmalı. Çünkü eşitliklere göre vektörlerin katsayılarının toplamı 0’a eşittir. O halde bu vektörler lineer bağımlıdır.

Başka bir örnek yapalım:

****

Çözüme geçelim:

Lineer bağımlı olduğuna göre vektörlerin katsayıların toplamı 0 olmalı

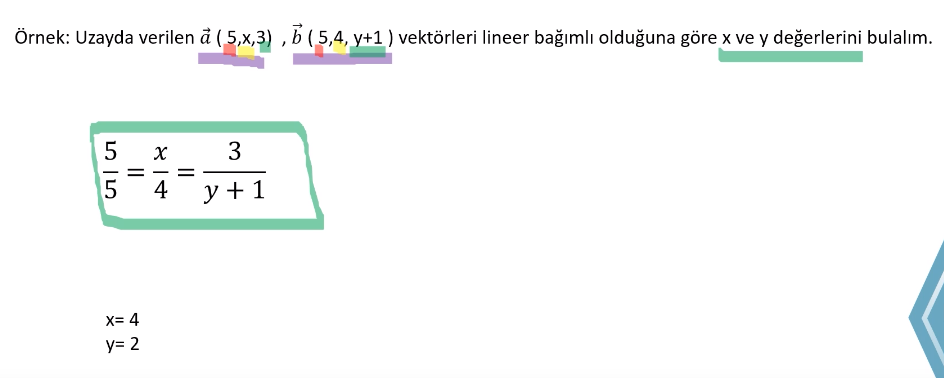
YA DA

katsayılar birbirine eşit olmalı

VEYA

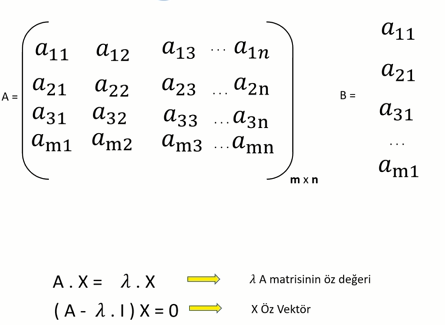
katsayılar oransal olarak birbirine eşit olmalı.

(matematiksel olarak son iki cümleyi söyleyebiliriz)

****

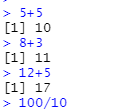
**4.21\_Öz Vektör**

Bir matrisin öz değerlerinin çarpımı o matrisin determinantını vermektedir.

****

**5\_R PROGRAMLAMA TEMELLERİ**

Temel işlemler:



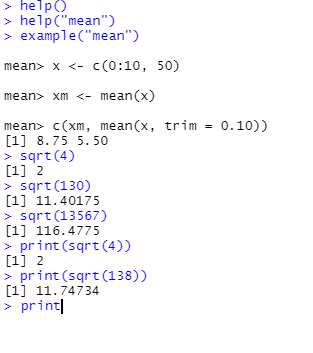
Atama işlemi:



ya da bu şekilde atama yapabiliriz.



NOT: CTRL+L ile Console temizleyebiliriz.

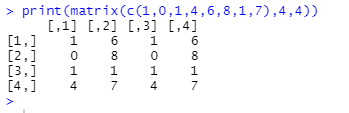


Şimdide bir matris oluşturalım

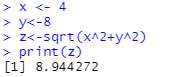
Değerleri yazdık ve 2x2 boyutunda bir matris olduğunu belirttik



şimdi ise 4x4 lük bir matris girelim







Daha önce ki tanımladığımız değikenleri ls() komutu ile gösterebiliyoruz:



Şimdi buradan bir tane değiş silelim:

>rm(a)

görüdüğü gibi değişkenler arasında artık a silindi.

> ls()

[1] "b" "c" "d" "e" "x" "xm" "y" "z"

iki değişkeni silelim:

> rm(b,d)

> ls()

[1] "c" "e" "x" "xm" "y" "z"

Mode ile değişkenlerimizin türünü öğrenebiliriz.

> mode(e)

[1] "numeric"

Şimdi ise iki adet vektör tanımlayalım ve iki vektorü birleştirelim



X değerinin ortlamasını hesaplayalım:

> x=c(1,2,3,6,8,9)

> mean(x)

[1] 4.833333

X in standart sapmasını alalım:

> sd(x)

[1] 3.311596

medyan ve varyansını hesaplayalım:

> median(x)

[1] 4.5

> var(x)

[1] 10.96667

1 den 10 a kadar dizi oluşturma:

> 1:10

[1] 1 2 3 4 5 6 7 8 9 10

seq ile 6 dan başlayıp 30 a kadar 2 artarak sayı listeleyelim:

> seq (from=6, to=30, by=2)

[1] 6 8 10 12 14 16 18 20 22 24 26 28 30

8 sayısından 10 kere yazmasını isteyelim:

> rep(8, times=10)

[1] 8 8 8 8 8 8 8 8 8 8

q değişkeninimizin 4 eşip olup olmadığını sorgulayalım:(True-False değeri döndürür.)

> q=4

> q==4

[1] TRUE

s değişkenimiz 12 ye eşit olup olmadığını sorgulayalım:

> s=12

> s !=12

[1] FALSE

a ve b dizisi tanımlayalım ve burdaki elelamların aynı olup olmadığını sorgulayalım

> a = c(2,5,7)

> b = c(3,5,7)

> a == b

a dizisinin elemanları arasında 3 var mı ?

> any(a==3)

[1] FALSE

a dizisinin elemanları arasında 3 var mı ?

> any(b==3)

[1] TRUE

"a dizisi b dizisinin elemanlarına eşit değil mi ?" sorusu soralım:

> a != b

[1] TRUE FALSE FALSE

a dizinin hepsinin 5 e eşit olup olmadığını sorgulayalım:

> all(a==5)

[1] FALSE