

Iris Dataset Analysis

210601036

Büşra Kurun

Veri Bilimi ve Makine
Öğrenmesine Giriş Dersi Vize
Ödevi



Kütüphaneleri Çağırma

- İhtiyacımız doğrultusunda kullanacağımız kütüphaneleri yükleyerek başlıyorum.
- Öncelikle pandas, numpy ve seaborn kütüphanelerini çağırarak başlıyoruz.
- Görünen uyarı mesajlarını filtrelemek için de bir kod parçası yazıyorum. Bu sayede, kodun okunabilirliği artacak ve gereksiz uyarılar ekrana yazdırılmayacak.

Veri Setini Çağırma ve Veri Setine Genel Bakış

- Veri setini kaggleden buldum ve bir veri çerçevesi haline getirerek `df_iris` değişkenine atadım.
- Ardından da ilk 5 elemanını görebilmek adına `head()` fonksiyonunu kullandım.
- `Head()` fonksiyonu; bir veri kümesinin başlangıçtaki satırlarını (varsayılan olarak ilk 5 satır) gösteren bir Pandas DataFrame metodudur. Veri kümesindeki verilerin yapısını, içeriğini ve düzenini hızlı bir şekilde anlamak için sıkça kullanılır. Başlangıçtaki kaç elemanı görüntülemek istiyorsak parantez içine o sayıyı yazmamız yeterli olacaktır.
- Son 5 elemanı gözlemlemek istiyorsak da `tail()` fonksiyonunu kullanabiliriz.

- Veriseti incelendiğinde İris(Süsen) çiçeğinin yaprak uzunluk ve genişliği bilgilerinin yer aldığı görülmekte. Her bir satır, bir çiçeğe ait ölçüm değerlerini gösterir.
- Özellikler sırasıyla sepal-length (alt yaprak uzunluğu cm), sepal-with (alt yaprak genişliği cm), pedal-length (üst yaprak genişliği), pedal-width (üst yaprak uzunluğu).
- Son sütunda da görüleceği üzere sınıflarımız ise setosa, versicolor ve virginica.

In [4]:

```
df_iris.head()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa



- Veri çerçevesinin kaç öznitelik ve kaç gözlemden oluştuğunu gözlemek için ise “shape” kullanılabilir. iris verisetimiz 150 satır(gözlem) 5 sütundan(öznitelik) oluşmaktadır.
- "shape" fonksiyonunun yanı sıra “columns” ve “dtypes” fonksiyonları ile set hakkında daha fazla bilgiye ulaşabiliriz.
- sepal_length sepal_width petal_length ve petal_width değerlerinin "float64" ile ifade edildiği görülmekte. Yani nümerik ifadeler.
- Fakat species değerleri object veri türünde. Yani kategorik değişken.
- Ayrıca Pandas'ta bu üç bilginin hepsini hatta daha fazlasını içeren “info” fonksiyonu mevcut. Veri çerçevesindeki değişkenlerin türlerini ve bellek kullanımını info() metodu ile görüntüleyebiliriz.

```
In [8]:
```

```
df_iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 150 entries, 0 to 149
```

```
Data columns (total 5 columns):
```

#	Column	Non-Null Count	Dtype
0	sepal_length	150 non-null	float64
1	sepal_width	150 non-null	float64
2	petal_length	150 non-null	float64
3	petal_width	150 non-null	float64
4	species	150 non-null	object

```
dtypes: float64(4), object(1)
```

```
memory usage: 6.0+ KB
```

- Info fonksiyonunu kullanarak aşağıdaki bilgilere ulaşabiliriz:
- İnceleme sonucunda 5 ayrı sütun, 150 satırdan oluşan bir verisetiyle beraberiz. Sütun isimleri '_sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'. Ve her bir sütunun dtype'ı listelenmiş durumda.
- Verimizin hiçbir satırında **NULL** değer bulunmadığı da görülüyor. Ancak, verilerin cm cinsinden uzunluk olması gerektiği düşünüldüğünde, 0 cm veya negatif değerlerin girilmiş olması durumunda, burada eksik verilerin bulunabileceğini söyleyebiliriz. Veriseti incelerken bu durumları da göz önünde bulundurmamız gerekiyor. **NULL** değer bulunmadı diyerek incelemeye devam etmemeliyiz. Aykırı değerlere göz atarak kafamızdaki bu soru işaretlerini silebiliriz.

Temel İstatistik

- **Standart sapma** bir veri serisinin dağılımının ne kadar yaygın olduğunu ölçen bir istatistiksel kavramdır. Yani, verilerin ne kadar birbirinden farklı olduğunu ölçer. Standart sapma, verilerin ortalama değerinden ne kadar uzakta olduğunu gösterir.
- **Ortalama**, bir veri serisindeki tüm sayıların toplamının sayı adedine bölünmesi ile elde edilen bir istatistik değerdir. Yani, verilerin toplamının sayı adedine bölünmesiyle ortalama hesaplanır. Veri setindeki sayıların dağılımı hakkında genel bir fikir verir.
- **Varyans**, bir veri kümesindeki değerlerin ortalamadan ne kadar uzakta olduğunu ölçen bir istatistiksel terimdir. Varyans, her veri noktasının ortalamadan ne kadar farklı olduğunu hesaplar ve bu farkların karelerinin toplamının, veri sayısının bir eksiği ile bölünmesiyle elde edilir. Varyans, bir veri kümesinin dağılımının ne kadar homojen olduğunu ölçer. Yüksek varyans, verilerin ortalamadan daha uzak olduğu ve dağılımın daha heterojen olduğu anlamına gelirken, düşük varyans, verilerin ortalamaya yakın olduğu ve dağılımın daha homojen olduğu anlamına gelir.

- Bu 3 bilgiyi de describe() fonksiyonu sayesinde tek bir tabloda görüntüleyebiliriz.
- **Describe() fonksiyonu**, veri çerçevesindeki sayısal değişkenlerin temel istatistiksel özelliklerini gösteren bir özet istatistikleri tablosu oluşturur. Bu özellikler, değişkenlerin ortalaması, standart sapması, minimum ve maksimum değerleri, çeyreklik değerleri ve gözlem sayısıdır.

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

1. Sepal uzunluđu (sepal_length) için ortalama değeri 5.84 ve standart sapması 0.83'tür.
 2. Sepal genişliđi (sepal_width) için ortalama değeri 3.05 ve standart sapması 0.43'tür.
 3. Petal uzunluđu (petal_length) için ortalama değeri 3.76 ve standart sapması 1.76'dır.
 4. Petal genişliđi (petal_width) için ortalama değeri 1.19 ve standart sapması 0.76'dır.
- Ortalama ve standart sapma, bir değışkenin varyansı hakkında fikir verir. Standart sapma, ortalama etrafında verilerin ne kadar yayıldığını gösterir. Örneđin, petal uzunluđu değışkeni için standart sapma 1.76'dır, bu da verilerin ortalama etrafında oldukça yayıldığını ve yüksek bir varyansa sahip olduğunu gösterirken, sepal genişliđi değışkeni için standart sapma 0.43'tür ve verilerin daha az yayıldığını ve daha düşük bir varyansa sahip olduğunun bir göstergesidir.



Eksik Veri Kontrolü

In [10]:

```
df_iris.isnull().sum()
```

Out[10]:

sepal_length	0
sepal_width	0
petal_length	0
petal_width	0
species	0
dtype: int64	

- Veri çerçevesinde hangi öznitelikte kaç adet eksik değer olduğunu gözlemleyelim.
- Gözlemlendiği üzere de her satırda 150 tane boş olmayan satır bulunmakta. Bu da demek oluyor ki verimizde eksik değer yok. Bunun sağlamasını da **isnull()** ve **sum()** fonksiyonları yardımıyla görebiliriz.
- Elimizdeki iris verisetinde bu sorguları yaptığımızda her değişken için 0 değerini aldım. Yani dataframede boş değer bulunmuyor.

Veri Görselleştirm e

Korelasyon

- Korelasyon değişkenlerimiz arasındaki oran diyebiliriz. Oranlar -1 ve 1 arasında çıkar. -1 negatif ilişki , 1 pozitif ilişki ,0 ilişki yok demektir. İlişki 1e ne kadar yakınsa ilişki o kadar çoktur diyebiliriz.
- Korelasyonu bulmak için `corr()` fonksiyonunu kullanıyorum.
- Korelasyon matrisindeki her bir değişken kendisiyle olan korelasyonu 1'dir çünkü bir değişkenin kendisiyle arasındaki ilişki tam olarak pozitiftir ve korelasyon katsayısı 1'dir. Diğer değişkenlerle arasındaki korelasyon ise farklı olabilir. Korelasyon matrisinde köşegen her zaman 1'dir, çünkü bir değişkenin kendisiyle olan korelasyonu **her zaman mükemmeldir.**

Korelasyon Tablosu

- Örneğin, sepal_width'in de petal_length'in de değişkenleri arasındaki korelasyon katsayısı 1'dir, **çünkü her bir değişken kendisiyle olan korelasyonu 1'dir.**
- En güçlü pozitif ilişki "petal_length" ve "petal_width" arasında görülür. Tablo incelendiğinde bu oranın 0.962757 olduğu gözükmemekte. Yani 1'e oldukça yakın. Bu da, çiçeklerin taç yapraklarının uzunluğu arttıkça, taç yapraklarının genişliğinin de arttığını gösterir.

In [11]:

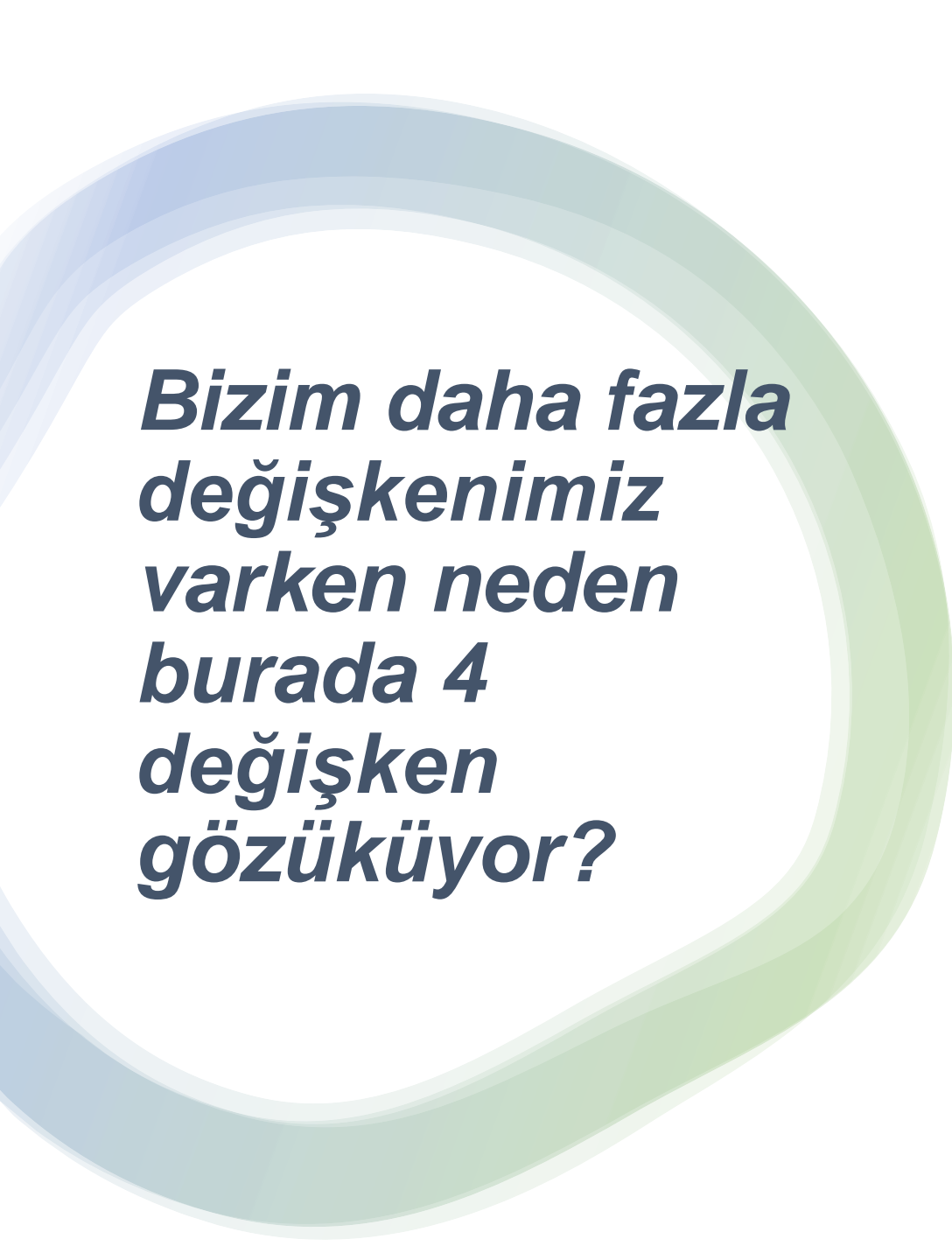
```
df_iris.corr()
```

Out[11]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

Ayrıca, "sepal_length" ve "petal_length" arasında da 0.871754 ile pozitif bir ilişki vardır, ancak "petal_length" ve "petal_width" arasındaki ilişki (0.96) daha güçlüdür.

Bununla birlikte, "sepal_length" ve "sepal_width" arasında neredeyse hiç ilişki yoktur (-0.109). Yani bitkinin çanak yaprağının boyuyla genişliğinin ilişkisi oldukça düşüktür.



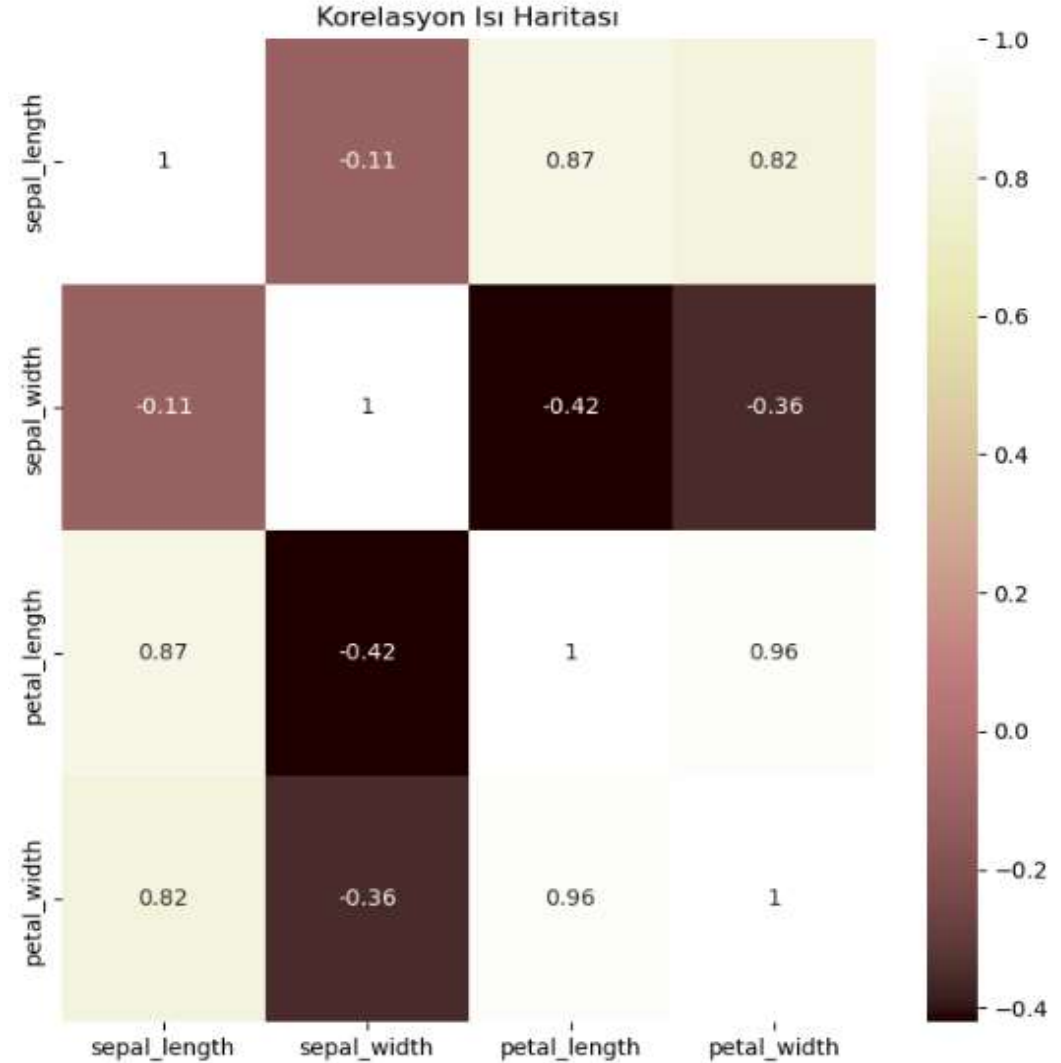
***Bizim daha fazla
değişkenimiz
varken neden
burada 4
değişken
gözüküyor?***

- Çünkü korelasyon **kategorik** değişkenlerde hesaplanamaz.
- Peki bu veri setindeki kategorik değişkenler ne? Bunu gözlemlemek için tekrardan veri setini çağırdım ve "species" değişkeni kategorik değişken olduğu için korelasyon grafiğinde yer almadığını gördüm.
- Korelasyonda da hesaplayabilmek için **kategorik değerler değil nümerik değer** olmalıdır. Korelasyonda yalnızca nümerik değerleri inceleyebiliriz.
- Korelasyon katsayılarını çok daha iyi incelemek için şimdi de ısı haritası kullanacağım.
- Isı haritasını çizebilmek için matplotlib kütüphanesini tanımlamamız gerekiyor. plt kısaltmasıyla matplotlib kütüphanesini import ediyorum.

Korelasyon Isı Haritası Çizdirme

- Renk skalasına göre; renk tonu açık olduğunda, değişkenler arasındaki ilişki o kadar yüksektir. Harita incelendiğinde, tekrardan en güçlü ilişkinin (1'den sonra) 0.96 ile "petal_length" ve "petal_width" arasında olduğu görülmektedir.

```
plt.figure(figsize = (8,8))  
plt.title("Korelasyon Isı Haritası");  
sns.heatmap(df_iris.corr() , cmap="pink" , annot=True);
```



Benzersiz Değerler

***Veri çerçevemizin
hedef değişkeninin
"variety" benzersiz
kaç adet değer
içeriyor?***

Bunun için Pandas kütüphanesinin "nunique()" fonksiyonu ile veri çerçevesinin belirtilen sütununun benzersiz değerlerinin sayısı bulunabilir.

Kodu yazıp çalıştırdıktan sonra 3 adet benzersiz değişken bulunduğunu gözlemliyorum.

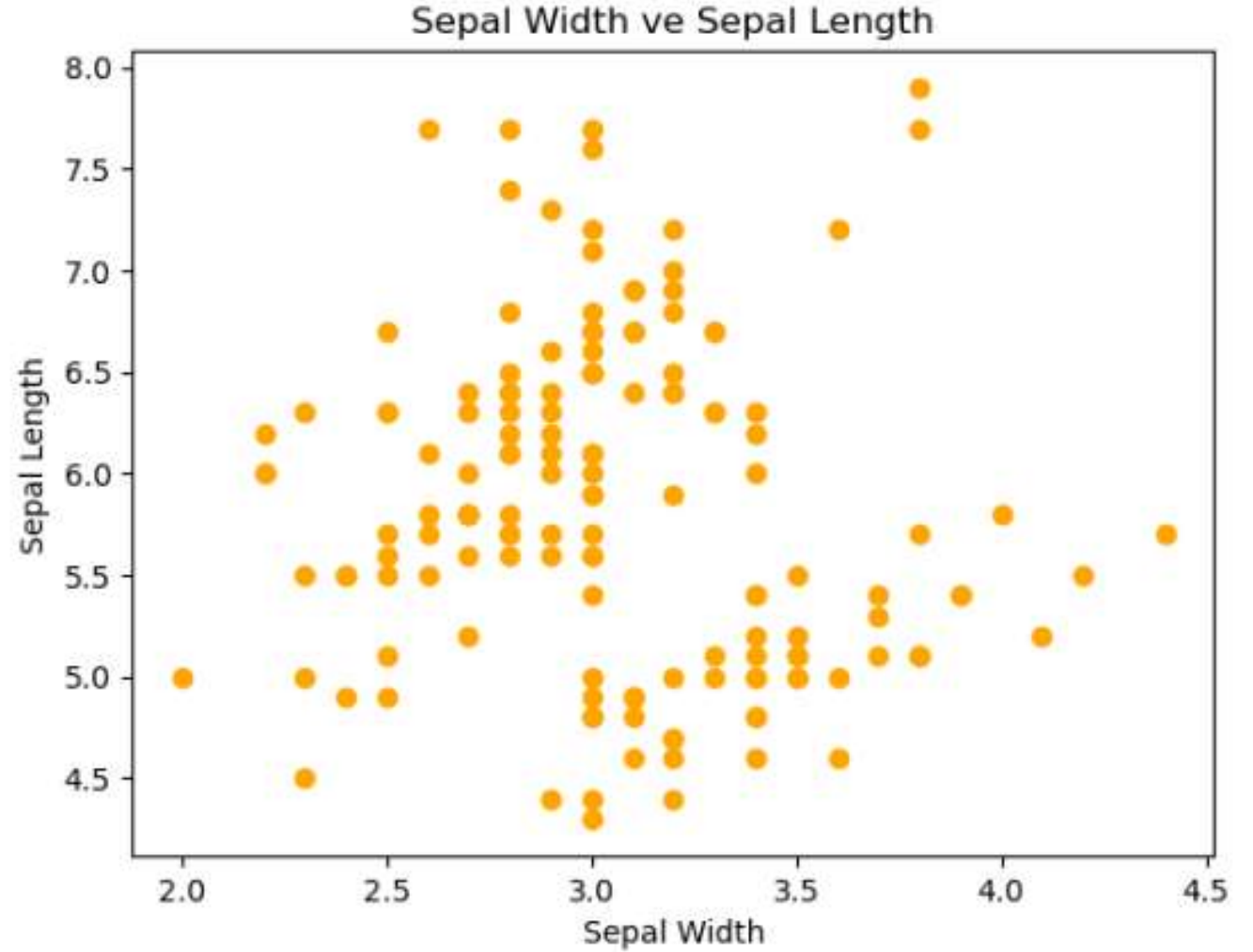
In [17]:

```
print(df_iris['species'].nunique() , "adet benzersiz değişkeni vardır.")
```

3 adet benzersiz değişkeni vardır.

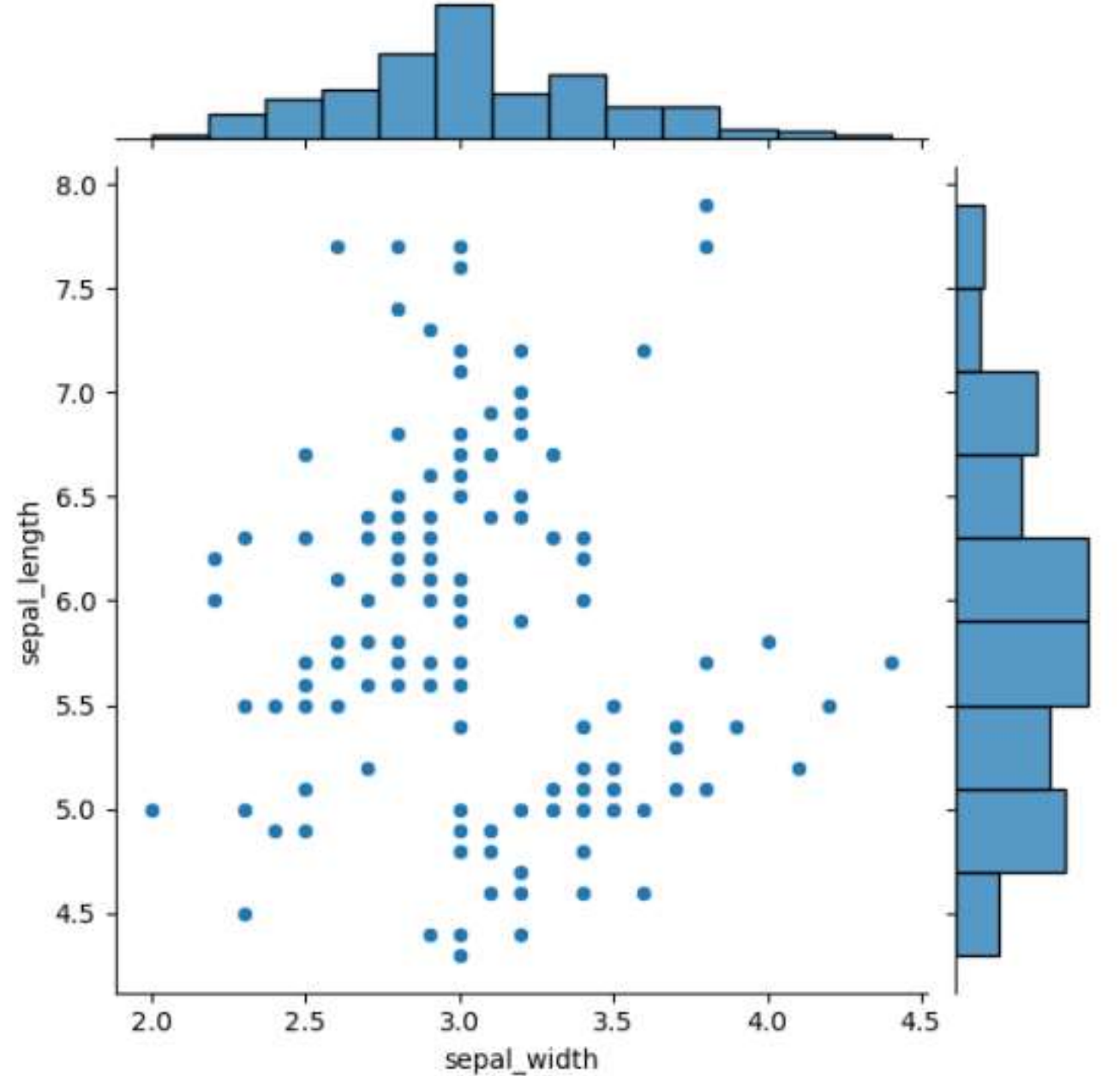
Veri Görselleştirmeye Devam Edelim...

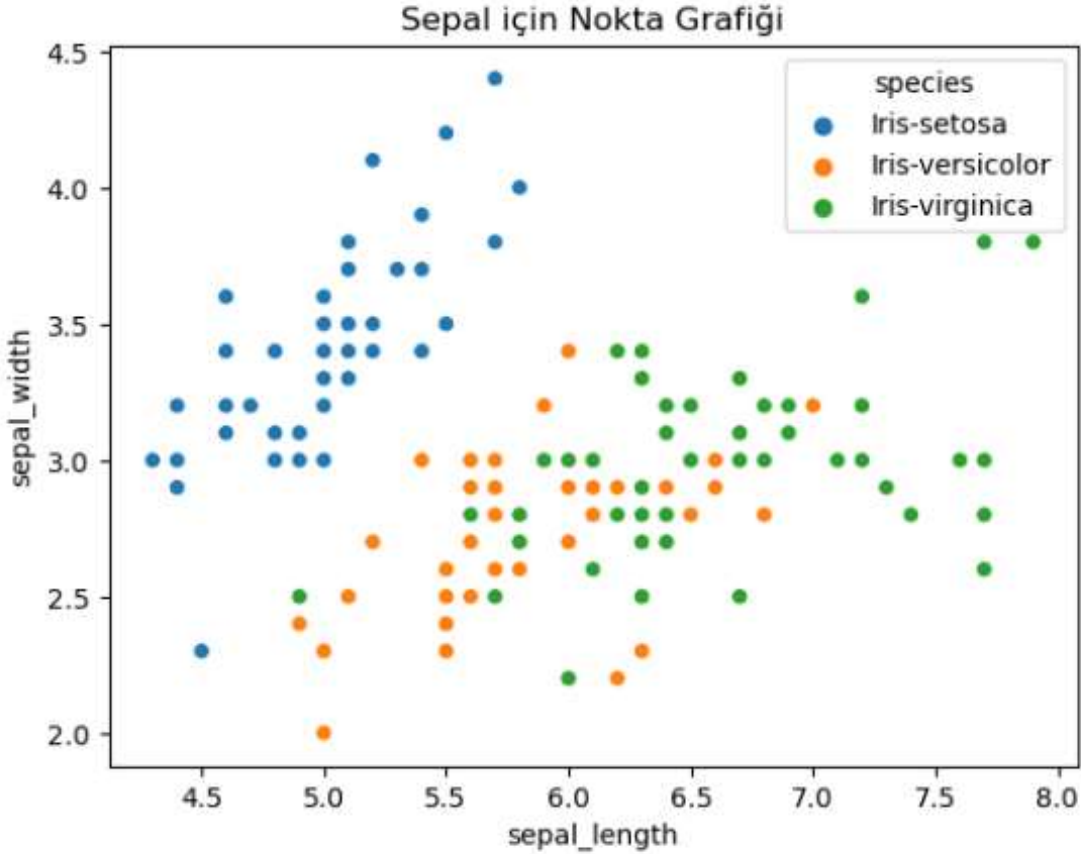
- Sepal Width değişkeninin değerleri 2.5-3.5 aralığında yoğun olarak gözükmemektedir. Length değişkeninin değerleri ise 5.5-7.0 arasında yoğunlaşmaktadır diyebiliriz.



Aynı iki veriyi daha farklı bir açıdan frekanslarıyla incelemek için jointplot kullanarak görselleştirmeyi deneyelim.

- Bu grafiğe bakarak bir önceki yorumumun kısmen doğru olduğunu görüyorum. Çünkü ilk grafikte length'in değerlerinin 4.5-5.0 arasında yoğunlaştığını görememiştim. jointplot grafiği yorum yapmak için daha mantıklı bir grafik.





Aynı iki veriyi scatterplot ile tekrardan görselleştirelim fakat bu sefer "variety" parametresi ile hedef değişkenine göre kırıralım.

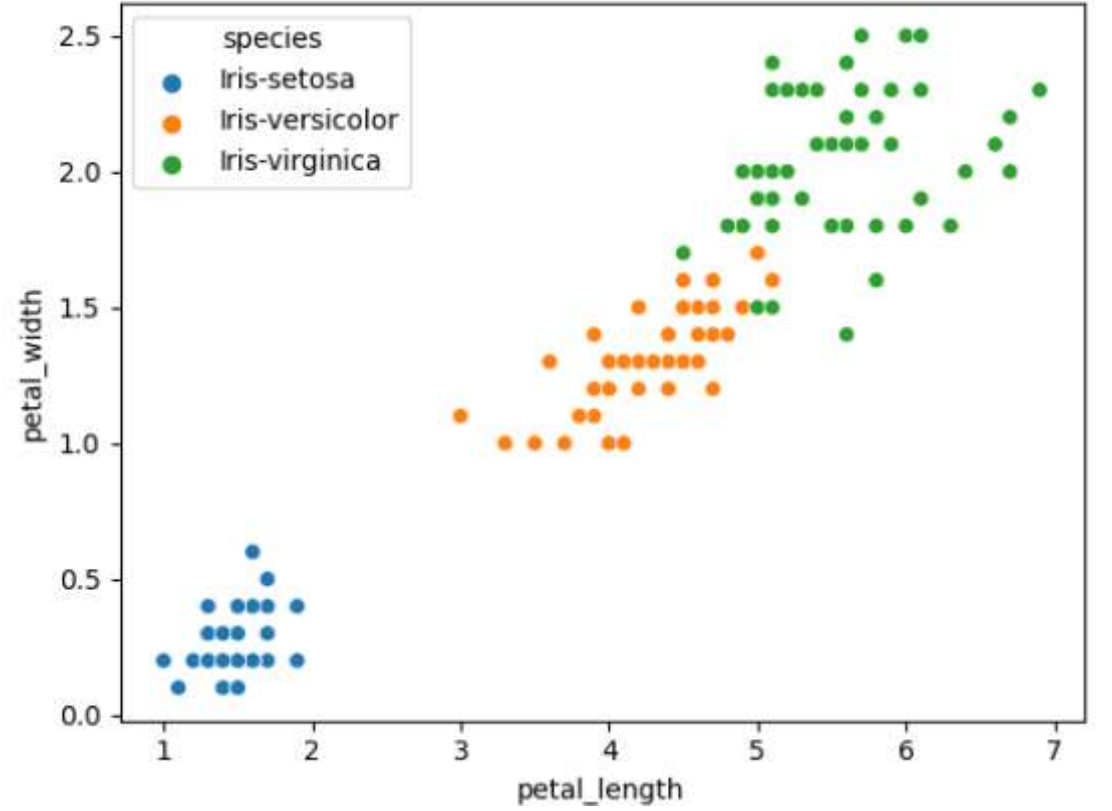
- Grafikte 3 farklı renk arasında sepal değişkenleriyle bir kümeleme yapılabildiği görülmektedir. Görüldüğü üzere lacivert renkli "Setosa" türü diğerlerinden daha iyi ayrılmaktadır, ancak turuncu renkli "Versicolor" ve yeşil renkli "Virginica" türleri birbirine daha çok benzemektedir. Daha çok iç içe geçmiş durumdadır. Bu nedenle, sadece sepal özellikleri kullanarak türleri kesin olarak ayırt etmek zordur. Başka özellikleri de analize dahil ederek yorumlamamız gerekir.

Sepal yorumunu yaptıktan sonra Petal deęiřkendeki durumu merak ettim. Hadi petal'e de bir bakalım...

- Mesela scatterplot fonksiyonu ile "petal_length" ve "petal_width" deęiřkenini incelersek durum nasıl olur?
- Bu scatterplot grafięi, petal uzunluęu (x eksen) ve petal geniřlięi (y eksen) deęiřkenlerinin çiçek t r ne (setosa, versicolor ve virginica) g re daęılımını g sterir. Her t r farklı bir renk ile temsil edilir. G rselden, virginica çiçeęinin genellikle dięer iki t rden daha b y k petallere sahip olduęu ve versicolor ve setosa'nın birbirine benzer boyutlara sahip oldukları g r lebilir. Ayrıca, setosa çiçeęi ile dięer iki t r arasında net bir ayırım g zlemlenebilir.
- Yani anlařılacaęı  zere Setosa çiçeęi gerek petal gerek ise sepal olarak dięer iki çiçek t r nden ayrılıyor.

In [21]:

```
sns.scatterplot(x="petal_length", y="petal_width", hue="species", data=df_iris);
```



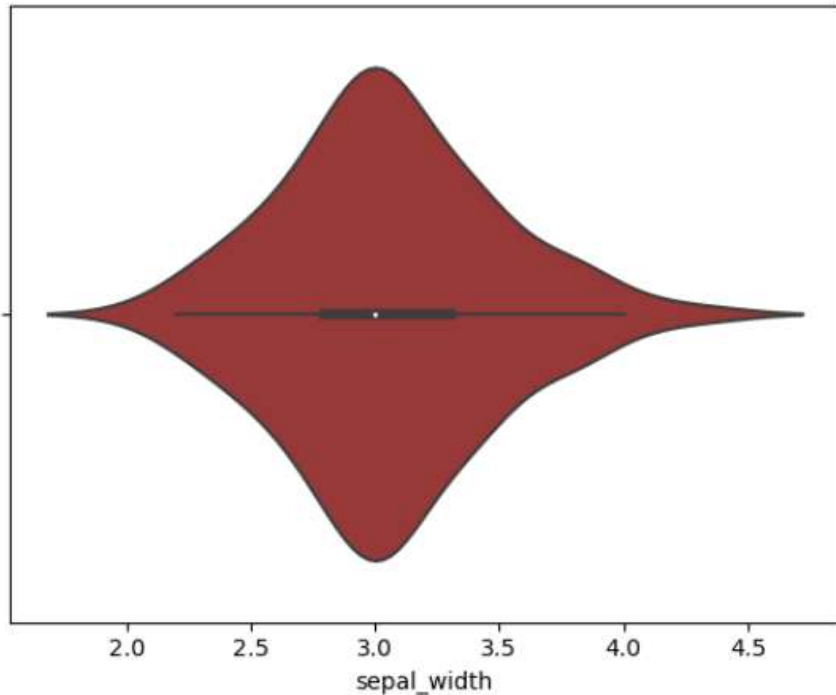
Veri Seti Dengeli Mi?

- `value_counts()` fonksiyonu ile veri setinin ne kadar dengeli dağıldığını öğrenebiliriz.
- `species` değişkenindeki her bir sınıfın veri çerçevesinde kaç kez tekrarlandığını sayar ve sonuçları sınıf adıyla birlikte gösterir.
- İnceleme sonucu `Setosa`, `Versicolor`, `Virginica` türlerinden de 50'şer adet olduğunu gözlemlemiş olduk. Bu da demek oluyor ki verisetimiz dengeli dağılmış.
- Eğer her sınıftan farklı sayıda örnek olsaydı, bu dengesiz bir dağılım olarak yorumlanabilirdi.

Keman Grafikleri

In [23]:

```
sns.violinplot(x="sepal_width", data=df_iris, c  
color="brown");
```

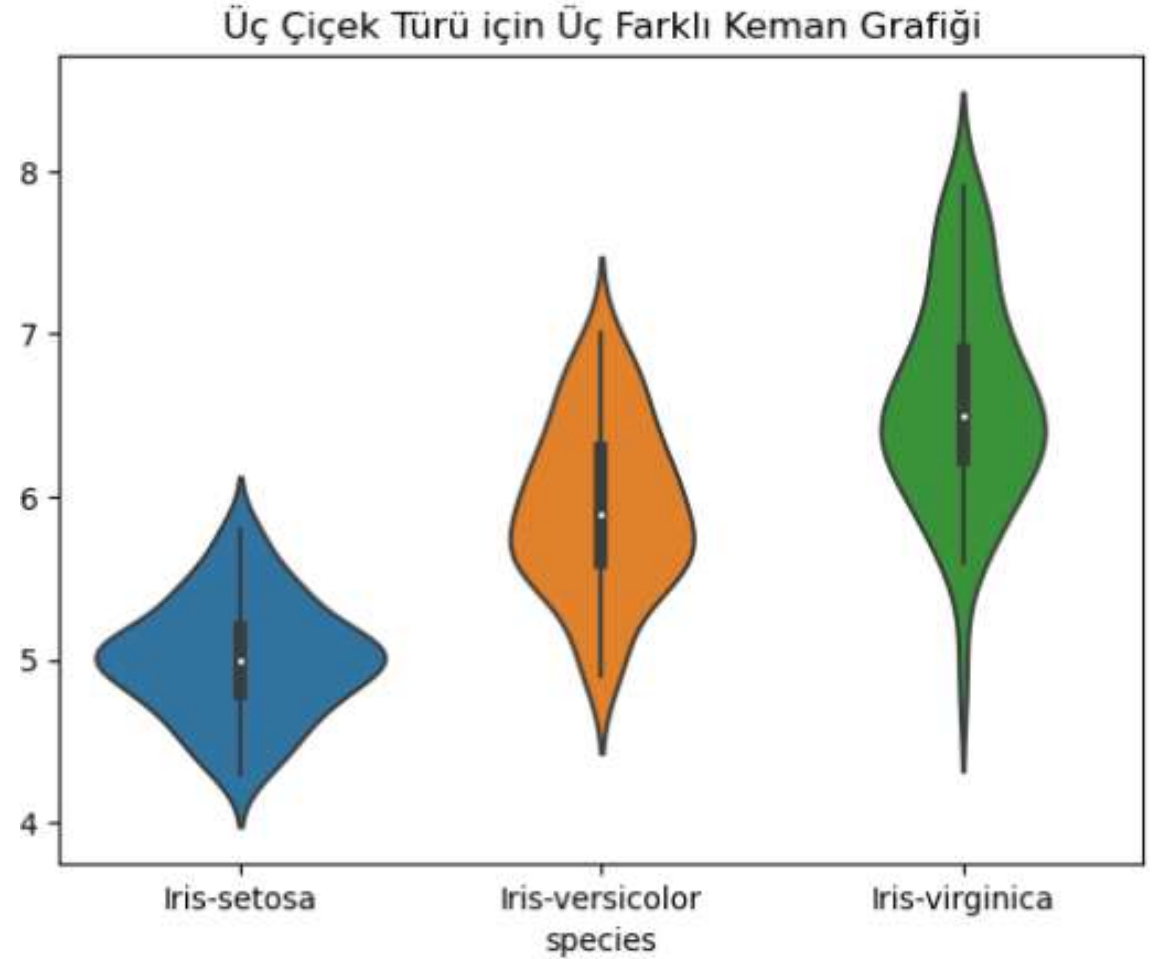


- ***Keman grafiği nedir?***

- Keman grafiği, veri dağılımının yoğunluk eğrisini gösteren ve aynı zamanda veri dağılımının simetrik mi yoksa çarpık mı olduğunu gösteren bir görselleştirme aracıdır.
- Sepal.width değişkeninin keman grafiğine bakarak, verinin normal bir dağılım göstermediği, hafif bir çarpıklık olduğu söylenebilir. Grafiğin sol tarafındaki kuyruk daha uzun ve çıkıntılıdır. Bu nedenle, sepal.width değişkeninin normal bir dağılım göstermediğini söyleyebiliriz.
- Grafiğin merkezindeki beyaz nokta, değişkenin ortalamasını gösterirken, çizginin uzunluğu, değişkenin dağılımını ifade eder. Burada sepal_width değişkeninin yoğunluklu olarak 2.5 ile 3.5 arasında dağıldığı ve geniş bir dağılıma sahip olduğu görülmektedir. Ayrıca, birkaç aykırı değer de görülmektedir.

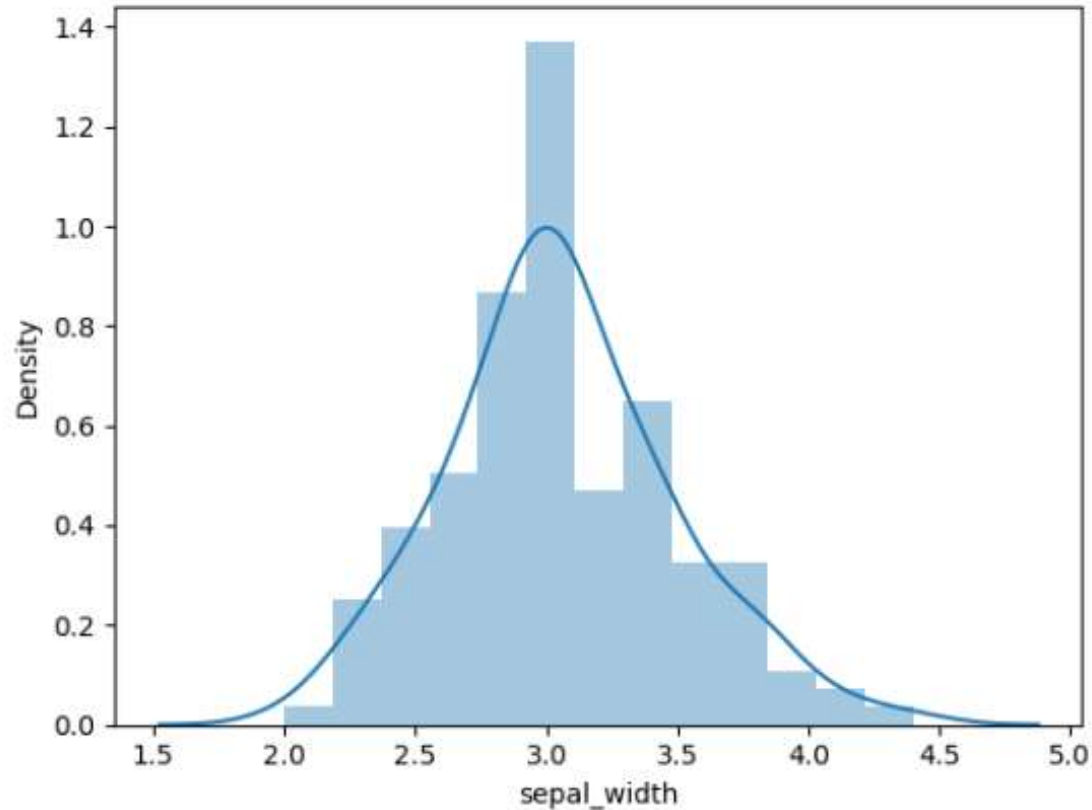
Üç çiçek türü için üç farklı keman grafiğini sepal.length değişkeninin dağılımı üzerine tek bir satır ile görselleştirelim.

- Yandaki keman grafiği gösteriyor ki Setosa çiçeklerinin sepal length değerleri birbirine daha yakın bir dağılım gösterirken, Virginica çiçeklerinin sepal length değerleri daha geniş bir aralıkta dağılıyor ve bu nedenle daha fazla değişkenlik gösteriyor diyebiliriz.



In [24]:

```
sns.distplot(df_iris['sepal_width']);
```

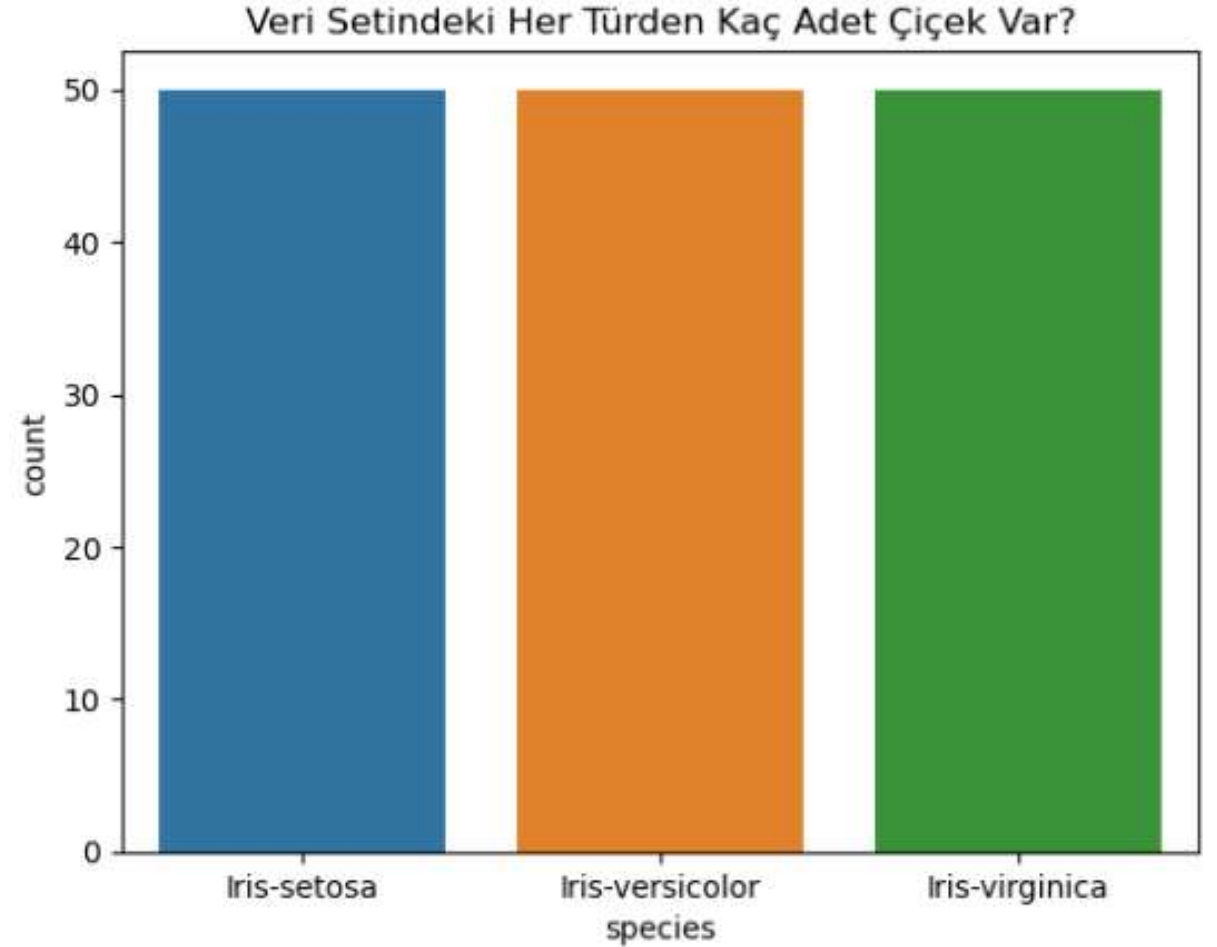


Daha iyi anlayabilmek için sepal.width üzerine bir distplot çizdirelim

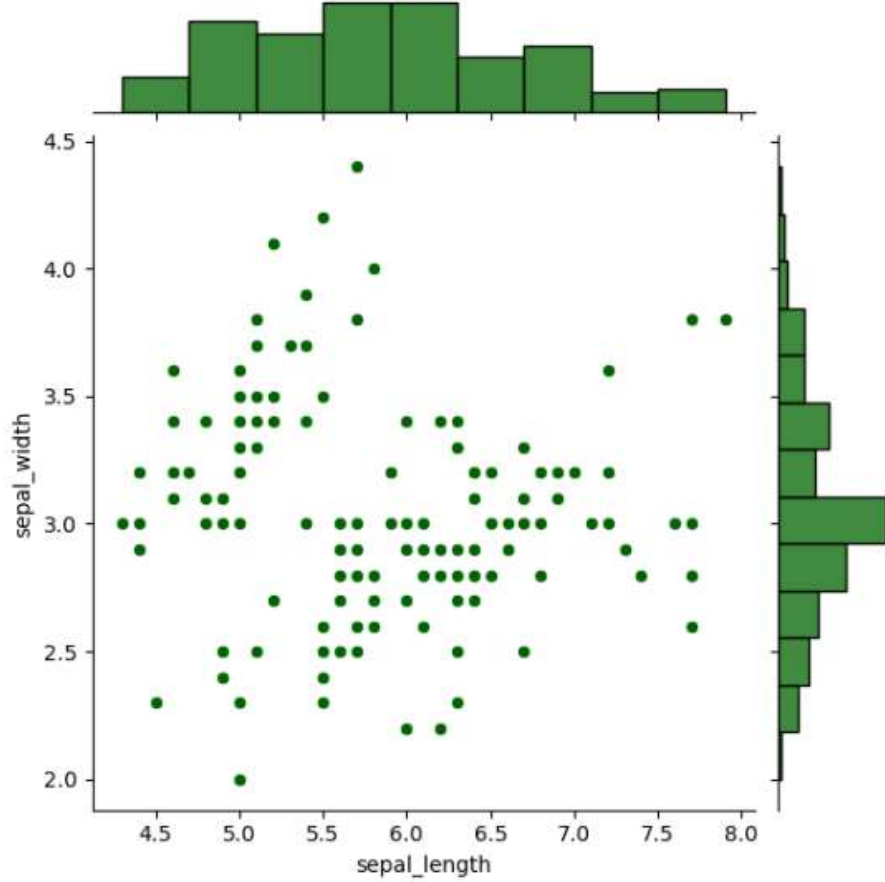
- Histogram, verilerin frekans dağılımını belirlerken, çizgi grafiği verilerin yoğunluğunu gösterir. Bu grafikte, sepal_width sütunundaki verilerin normal dağılıma yakın bir dağılım gösterdiği, ancak bazı aykırı değerlerin de olduğu görülebilir.

Veri setimiz hangi çiçek türünden kaç adet gözlem barındırıyor?

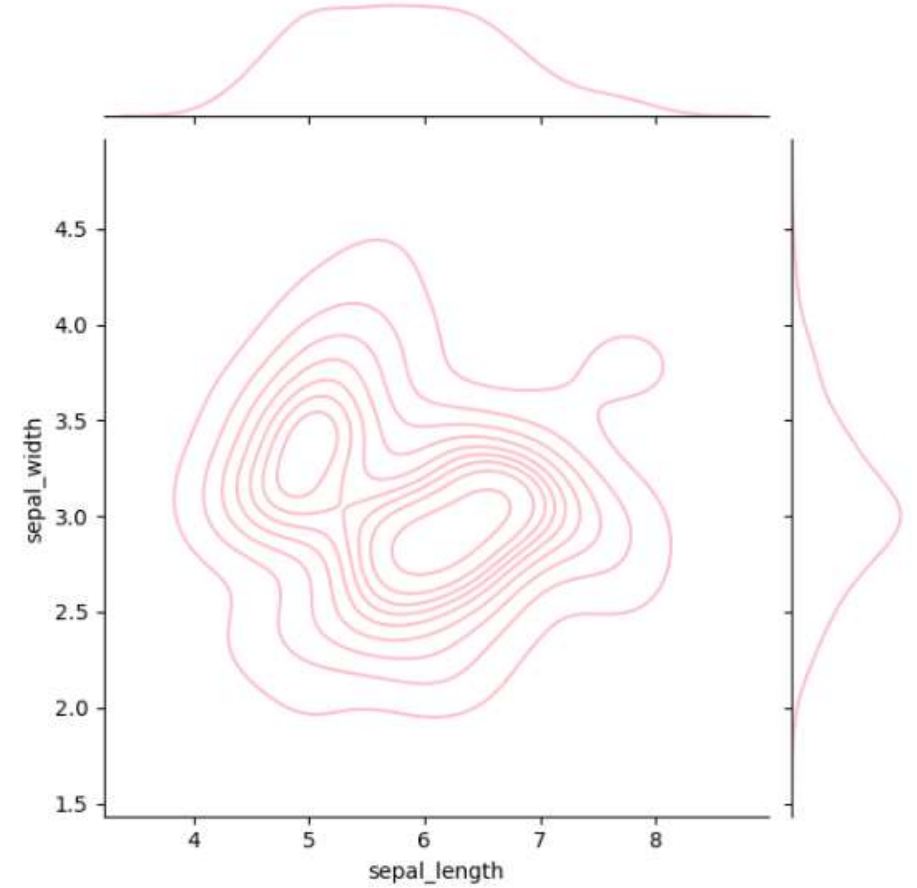
- 50 x 3 olduğunu ve dengeli olduğunu value_counts ile zaten görmüştüm, ancak bunu görsel olarak ifade etmek için sns.countplot() fonksiyonuna variety parametresini vereceğim. Görsel olarak da görmek benim için daha iyi olacaktır.



Sepal length ve sepal width deęişkenlerini sns.jointplot ile g rselleřtirdim.

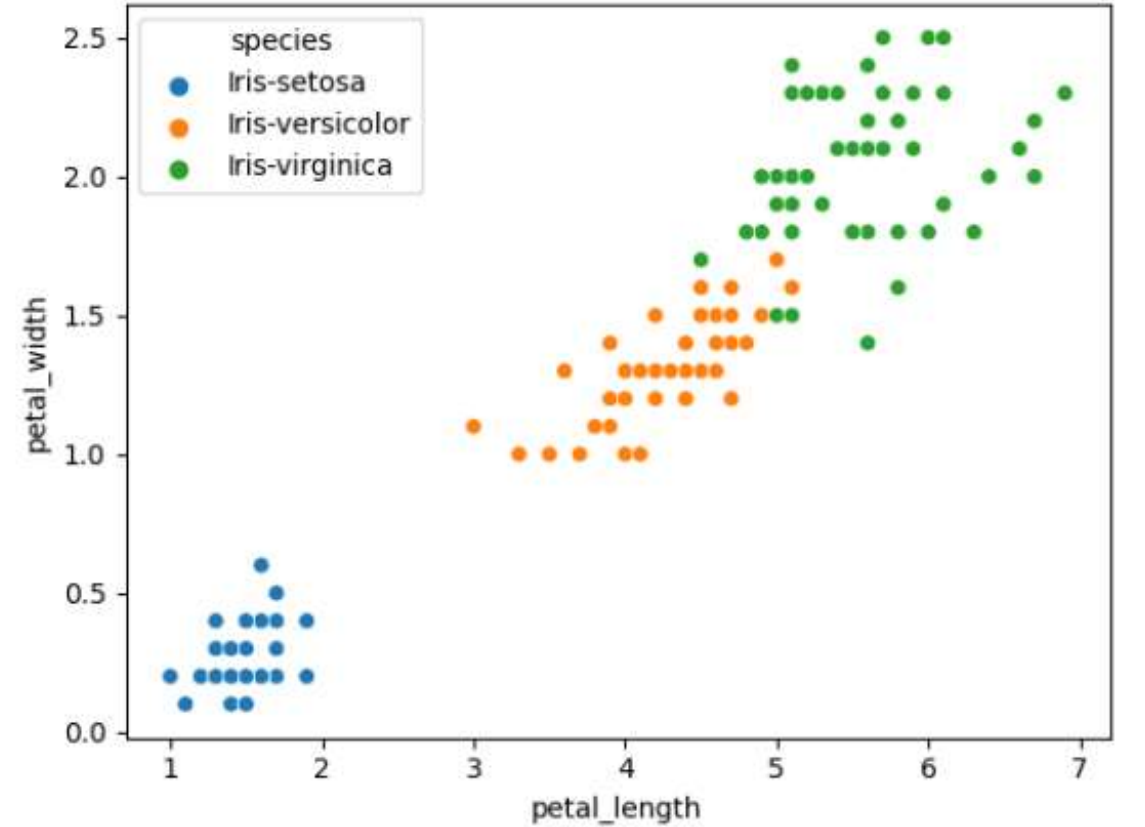


Yandaki grafięe kind = "kde" parametresini ekleyelim. B ylelikle daęılımın noktalı g sterimden  ıkıp yoęunluk odaklı bir g rselleřtirmeye d n řt ę n  g rm ř olacaęız.



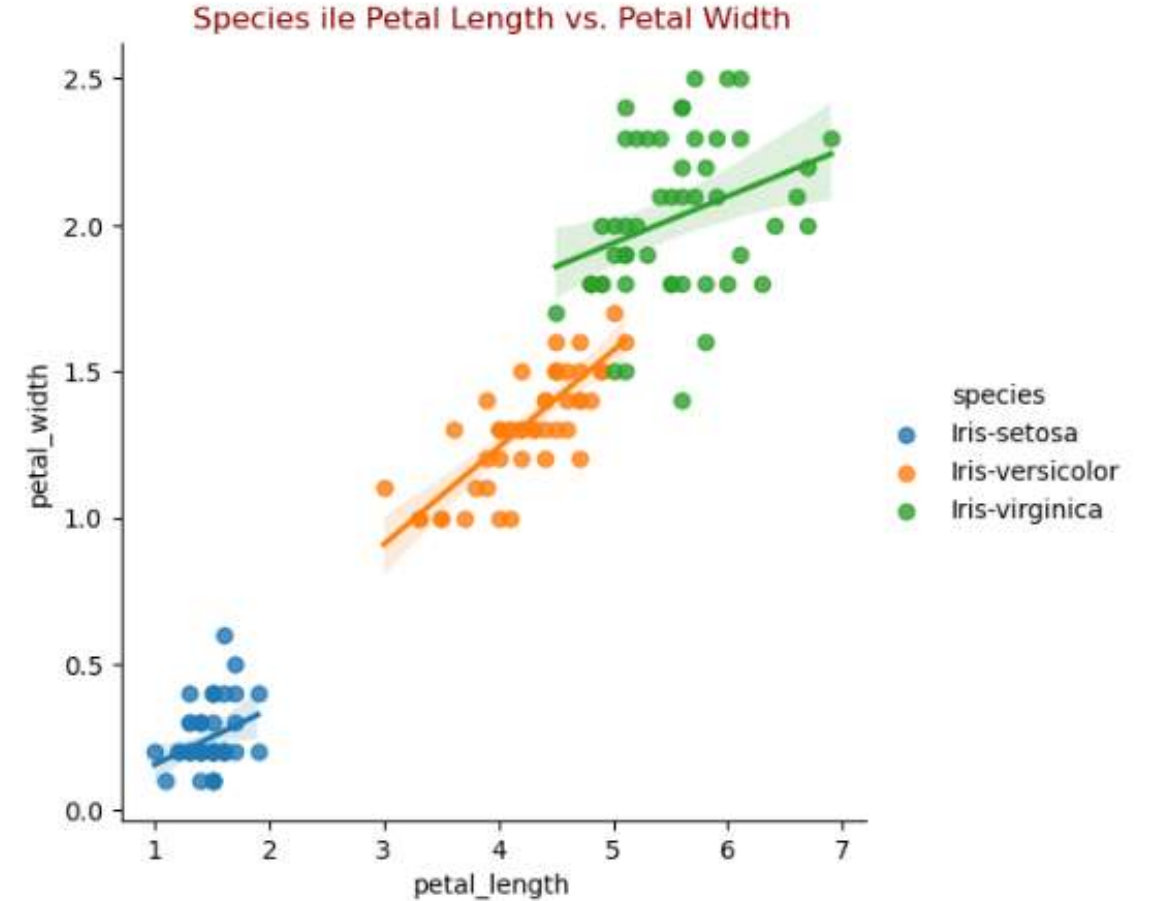
Petal length ve petal width deęiřkenlerinin nokta grafikle daęılımlarını inceleyelim.

- "petal_length" deęiřkeninin x-ekseninde, "petal_width" deęiřkeninin y-ekseninde olduęu bir scatterplot oluřturur.
- Bu grafięe gre; setosa tr dięer iki tre gre daha kk boyutlu ve daha dar yapraklara sahiptir ıkarımında bulunabiliriz.



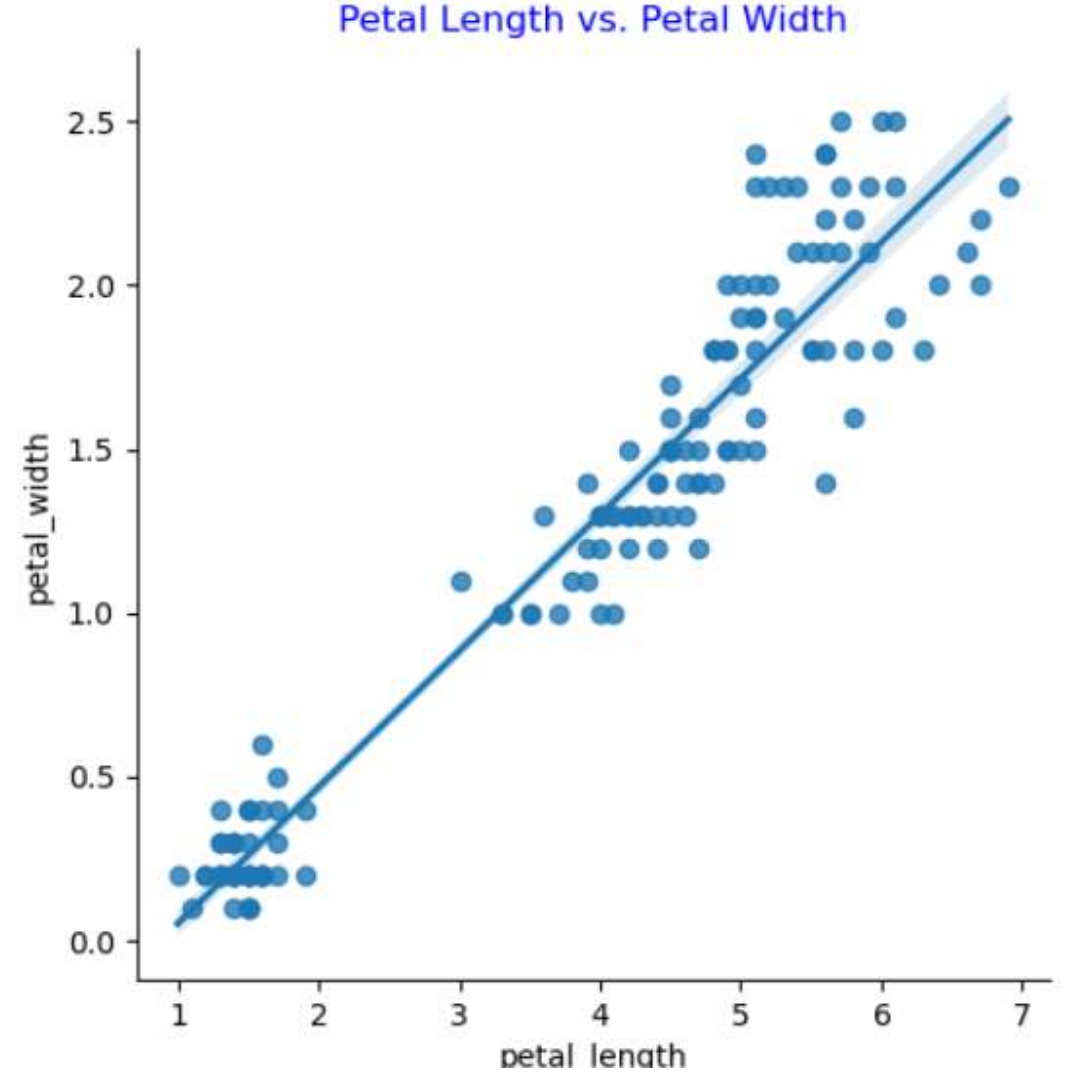
Petal length ile petal width arasında ne tür bir ilişki var ve bu ilişki güçlü müdür?

- Bu grafikte petal length ile petal width arasında pozitif bir ilişki olduğu görülmektedir. Yani bir çiçeğin petal length değeri arttıkça, petal width değeri de genellikle artmaktadır. Ayrıca, her bir çiçek türü için (setosa, versicolor ve virginica), bu ilişkinin farklı bir doğru üzerinde olduğu görülmektedir.
- Özellikle setosa çiçeklerinin (lacivert renkli noktalar) petal length ve petal width değerleri arasındaki ilişki, diğer iki çiçek türüne göre daha zayıf görünmektedir. Bunun nedeni, setosa çiçeklerinin diğer çiçek türlerine göre daha küçük boyutlara sahip olması olabilir. Tabii ki bu çıkarım kesinlikle doğrudur demek yanlış olur. Diğer durumlarla birlikte daha detaylı incelenmesi gerekir.
- Grafikteki çizginin oldukça dik olması, bu değişkenler arasındaki ilişkinin yüksek korelasyonlu olduğunu göstermektedir. Grafikte, özellikle Versicolor çiçeklerinin (turuncu renkli noktalar) arasındaki ilişki en yüksek korelasyonlu olanıdır.



Grafięi iek trlerine gre sınıflandırmadan tek bir ekilde de gsterebiliriz.

- Bu grafięe bakarak da petal uzunluęu ve geniřlięi arasında pozitif bir iliřki olduęunu grebiliyoruz. Veri noktaları izgiye yakın bir ekilde gruplanmıřtır, bu da yksek bir korelasyon olduęunu gstermektedir. Species deęiřkenine gre sınıflandırmadan tek tip olarak gsterilmesi, tm iek trleri arasındaki iliřkiyi daha net bir ekilde grmemizi saęlar.
- Bu sorunun yanıtını pekiřtirmek iin iki deęiřken arasında korelasyon katsayısını yazdıralım.
- Daha nceden de hatırlarsınız ki en byk korelasyon 0.96 ile petal_length ile petal_width arasındaydı. Ve yine 0.96 ıktı.
- Bu sonu, yksek bir pozitif iliřki olduęunu ve iki deęiřkenin birlikte arttıęını gsterir. Ayrıca bu sonu, daha nce yapılan grselleřtirme analizinde de grldę zere, petal length ve petal width arasındaki iliřkinin gl olduęunu doęrulamaktadır.



Peki bütün length değerlerini toplayıp veri setine total bir length değeri eklemek istersek? Bunu yapabilir miyiz? Evet. Peki nasıl?

- Petal Length ile Sepal Length değerlerini toplayarak yeni bir total length özneliği oluşturdum ve verisetinin sonuna total_length değişkeni olarak ekledim.
- Artık yeni tablomuz şekildeki gibi oldu.
- Önceden 5 sütundan oluşan tablo şuan 6 sütundan oluşuyor.

In [34]:

```
df_iris.assign(total_length = df_iris['petal_length'] + df_iris['sepal_length'])
```

Out[34]:

	sepal_length	sepal_width	petal_length	petal_width	species	total_length
0	5.1	3.5	1.4	0.2	Iris-setosa	6.5
1	4.9	3.0	1.4	0.2	Iris-setosa	6.3
2	4.7	3.2	1.3	0.2	Iris-setosa	6.0
3	4.6	3.1	1.5	0.2	Iris-setosa	6.1
4	5.0	3.6	1.4	0.2	Iris-setosa	6.4
...
145	6.7	3.0	5.2	2.3	Iris-virginica	11.9
146	6.3	2.5	5.0	1.9	Iris-virginica	11.3
147	6.5	3.0	5.2	2.0	Iris-virginica	11.7
148	6.2	3.4	5.4	2.3	Iris-virginica	11.6
149	5.9	3.0	5.1	1.8	Iris-virginica	11.0

150 rows × 6 columns

Total Length değerini yakından inceleyelim.

Ortalama ve standart sapmasını hesaplayıp yorumlayalım.

Bir değişkenin ortalamasını bulabilmek için `mean()` fonksiyonu, standart sapmasını bulmak için ise `std()` fonksiyonu kullanılır.

Ortalama uzunluk 9.6 olarak bulunmuştur. Standart sapma ise 2.5'dir.

Ortalama 9.6, `total_length` değişkeninin tüm örneklem verilerinin toplamının örneklem sayısına bölünmesiyle hesaplanan bir değerdir. Standart sapma ise verilerin ne kadar yayıldığını gösteren bir ölçüttür. Bu durumda, standart sapmanın 2.5 olması, tüm çiçeklerin uzunluklarının ortalamadan ne kadar **farklılaştığını** gösterir. Yani, çiçeklerin uzunlukları ortalama uzunluktan yaklaşık 2.5 birim kadar sapma gösteriyor.

In [36]:

```
total_length_mean= df_iris["total_length"].mean()  
print("Ortalama Değer: ", total_length_mean)
```

Ortalama Değer: 9.602

In [37]:

```
total_length_std= df_iris["total_length"].std()  
print("Standart Sapma: ", total_length_std)
```

Standart Sapma: 2.5191739884121973

Sepal length'in maksimum değerini yazdıralım.

In [38]:

```
sepal_length_max= df_iris["sepal_length"].max  
(  
print( "Sepal Length'in Maksimum Değeri",sepal  
_length_max)
```

Sepal Length'in Maksimum Değeri 7.9

Sepal length'i 5.5'den büyük ve türü setosa olan gözlemleri yazdıralım.

```
In [40]: df_iris.loc[(df_iris['petal_length'] < 5) & (df_iris['species'] == 'virginica'), ['sepal_length', 'sepal_width']]
```

Out[40]:

sepal_length	sepal_width
--------------	-------------

Petal length'i 5'den küçük ve türü virginica olan gözlemlerin sadece sepal length ve sepal width değişkenlerini ve değerlerini yazdıralım.

```
In [39]: df_iris[(df_iris["sepal_length"] > 5.5) & (df_iris["species"] == "setosa")]
```

Out[39]:

sepal_length	sepal_width	petal_length	petal_width	species	total_length
--------------	-------------	--------------	-------------	---------	--------------

Her iki sorguda da seçilen koşullara uyan herhangi bir gözlem bulunamadı, yani birinci örnekte sepal length'i 5.5'den büyük ve türü setosa olan bir gözlem yoktu. İkinci sorguda ise petal length'i 5'den küçük ve türü virginica olan bir gözlem yoktu.

Hedef değişkenimiz variety'e göre bir grupta işlemi yapalım değişken değerlerimizin ortalamasını görüntüleyelim.

- Bu işlem için groupby() fonksiyonunu kullanabiliriz.

- Iris setosa, diğer türlerden daha küçük çiçekleri olan bir türdür. Diğer taraftan, Iris virginica, en büyük çiçeklere sahip olan türdür. Toplam uzunluk sütunu, çiçeklerin boyutunu tam olarak ifade etmek için hesaplanan bir özelliktir. Bu tablodaki veriler, türler arasındaki farklılıkları ve benzerlikleri anlamamıza yardımcı olabilir.

```
In [41]: df_iris.groupby('species').mean()
```

```
Out[41]:
```

	sepal_length	sepal_width	petal_length	petal_width	total_length
species					
Iris-setosa	5.006	3.418	1.464	0.244	6.470
Iris-versicolor	5.936	2.770	4.260	1.326	10.196
Iris-virginica	6.588	2.974	5.552	2.026	12.140

In [42]:

```
df_iris.groupby('species')['petal_length'].std()
```


Out[42]:

```
species
Iris-setosa      0.173511
Iris-versicolor  0.469911
Iris-virginica   0.551895
Name: petal_length, dtype: float64
```

Gruplandırma yaparak sadece petal.length değişkenimizin standart sapma değerlerini yazdıralım.

Yukarıdaki kodda ilk satırda groupby() fonksiyonunu kullanarak species değişkenine göre gruplama yapıyoruz ve sadece petal_length değişkenini seçiyoruz. Ardından da std() ile standart sapmayı hesaplıyoruz.

Setosa türünde petal length değişkeninin standart sapması en düşük, yani veriler arasındaki fark çok azdır. Iris-virginica türünde ise petal length değişkeninin standart sapması en yüksektir, yani veriler arasındaki fark daha fazladır. Bu da gösteriyor ki farklı türlerin özellikleri birbirinden farklıdır.

An abstract graphic on the left side of the slide. It features a white background with various colorful shapes and patterns. There's a blue oval at the top left, a large teal shape with a dotted pattern on the right, and a yellow shape with a cross pattern at the bottom left. Several small black squiggly lines are scattered throughout the white space.

İncelediğim bu verisetinin linkini ve slayttaki Kernel'a Kaggle hesabıma girerek ulaşabilirsiniz. Diğer sosyal medya linklerimi de aşağıya ekliyorum. Dilerseniz inceleyebilirsiniz.

Kaggledaki diğer veriseti incelemelerime de bakmayı unutmayın.

- <https://www.kaggle.com/brakurun>
- <https://medium.com/@busraakurun>
- <https://github.com/busrakurunceng>

Dinlediğiniz için teşekkür ederim.