Veri Görselleştirme Eğitimi

Kursa Başlamadan Önce

- Bu kurs tamamlandığı takdirde giriş düzeyi veri görselleştirme ve veri analizine temel oluşturabilecek genel bilgileri edinmiş olacaksınız
- Spesifik konular anlaşılması zor ve kişide ders esnasında mantığın oturması kolay olmayacağından bol bol bireysel pratik gerekmektedir
- Slaytlar yazılara boğulmadan görsellerle anlatılacaktır. Bu yüzden ders esnasında not tutulması **son derece** önemlidir
- Konu başlıkları temel düzey algoritmalar için yeterli olduğundan başlıklar araştırılmalı, bol bol uygulama ve teorik bilgiler içeren sitelerde araştırma yapılmalıdır

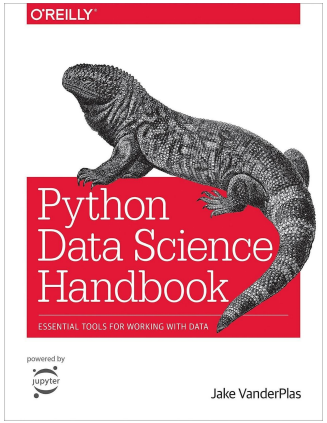
Bu Eğitimde Neler Öğreneceksiniz

1. Veri nedir, hayatımıza nasıl etki etmektedir ?
2. Veri Görselleştirme için gereklilikler ve ek yetenekler nelerdir?
3. Python'da En Popüler Veri Görselleştirme Kütüphaneleri ve Teknikleri
4. En Popüler Veri Görselleştirme Tipleri
5. Her görselleştirme tipi için basit ve neredeyse tek satırda pratik örnekler
6. **Verinin okunmasından, PDF rapor haline gelene kadar sürecini gösteren uçtan uca bir proje!**

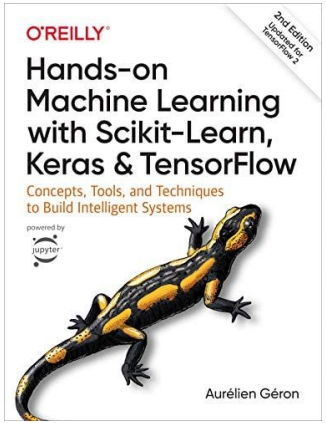
Ömer Cengiz

Github: omercengiz
Linkedin: omercengiz96
Twitter: omerrcengizz

Kaynaklar



Python Data Science Handbook
Jake VanderPlas



Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow
Aurelien Geron

Stanford
University

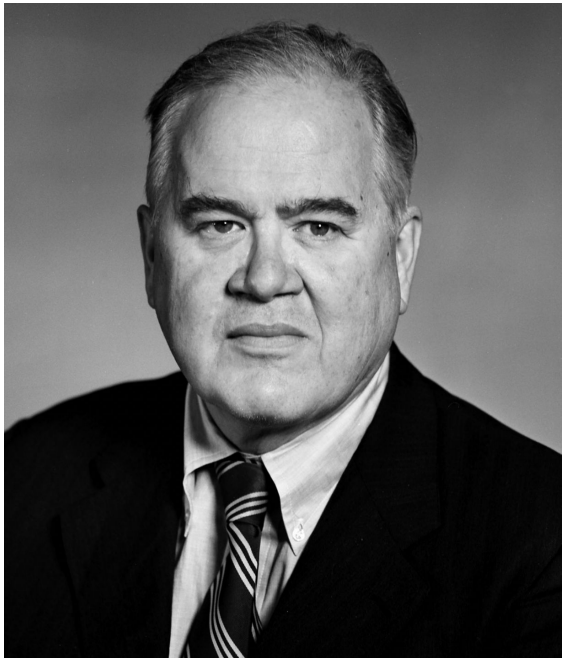
Stanford University stanford.edu/~shervine/
Shervine Amidi

Tukey

John Tukey, istatistik alanında eser vermiş ve çalışmalar yapmış büyük isimlerden biridir, aynı zamanda **Keşifçi Veri Analizi'nin (EDA)** mucididir.

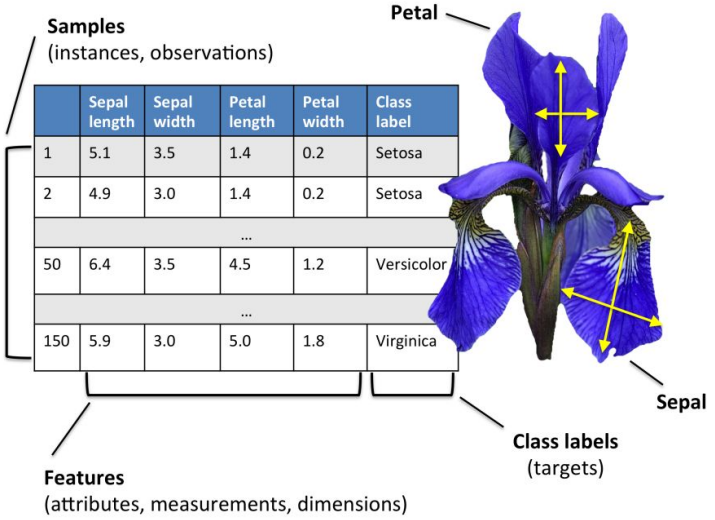
Tukey, mühendislik, matematik, istatistik ve bilimin çeşitli alanlarına katkıda bulunmuştur. Bunların çoğu, 1984'te ciltleri yayınlanmaya başlayan Toplu Eserler'de kayıtlıdır.

- Time Series (2 cilt)
- Philosophy and Principles of Data Analysis (2 cilt)
- Graphics
- More Mathematics
- Factorial and ANOVA
- Multiple Comparisons



Veri Nedir?

- Bilginin yapılandırma ile kayıt altına alınıp, kolay analiz edilebilmesi için bir araya getirilmesidir
- Bir veya birden fazla bilgiden oluşan kümedir
- Genellikle araştırma, gözlem, deney, sayım, ölçüm yoluyla elde edilir
- Yaş, isim, telefon numarası, bir toplama işleminin sonucu ya da sınıfın yaş ortalaması birer veridir



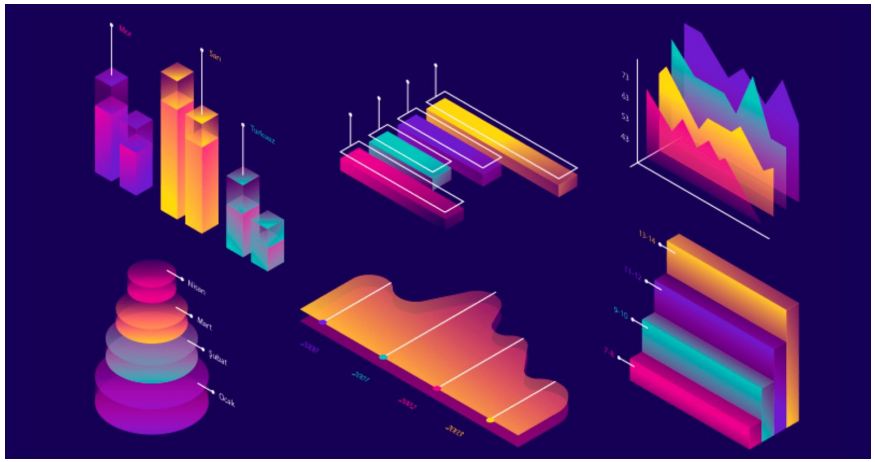
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Veri Görselleştirme Nedir?

Veri görselleştirme, verilerin nicel tanımlamalarına ve tahminlerine odaklanan uygulamalı istatistik ve makine öğreniminde önemli becerilerden biridir.

Bir veri kümesini keşfederken aşağıdakileri gözlemlemek amaçlanır:

- Patternler
- Aykırı değerler
- Trendler
- Dağılımlar



Veri görselleştirme, ilişkileri ifade etmek ve basitçe göstermek için kullanılır.

Karşılaştığınız çoğu veri hakkında hızlı bir şekilde niteliksel bir anlayış elde edebilirsiniz.

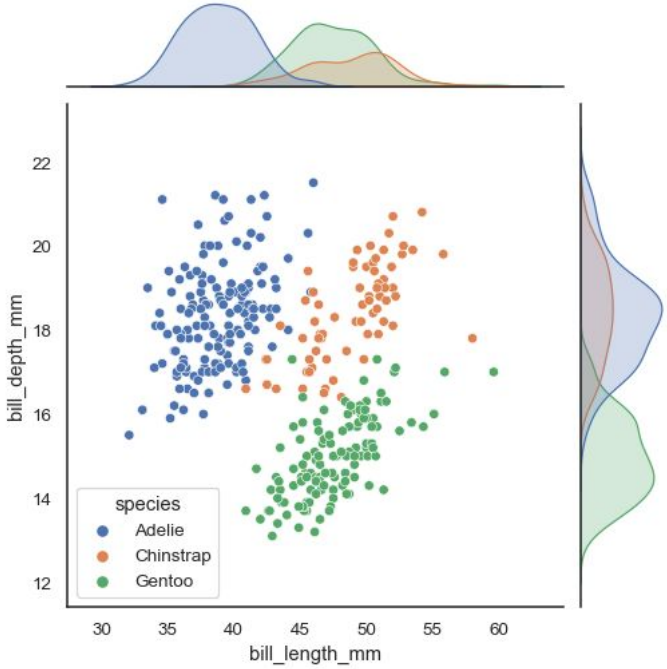
Veri Görselleştirmesinin Avantajları

Daha İyi Analiz: Veri tablolarında gözden kaçan unsurları, trendleri ve dağılımları incelemede oldukça etkilidir.

Hızlı Aksiyon: Veri görselleştirmeleri, verilerin hızlı bir şekilde anlamlandırılmasına ve yorumlanmasına olanak sağlar.

Örüntülerin Tanımlanması: Büyük miktarlardaki karmaşık veriler, görselleştirildiklerinde içgörüler için kolaylık sağlayabilir ve veriler arasındaki ilişkilerin bulunmasına olanak tanır.

Hata bulma: Görselleştirmek, verilerdeki hataları hızlıca belirlemenize yardımcı olur. Veriler yanlış örnekler bulundurma eğilimindeyse, görselleştirilerek hızlıca tespit edilebilir.



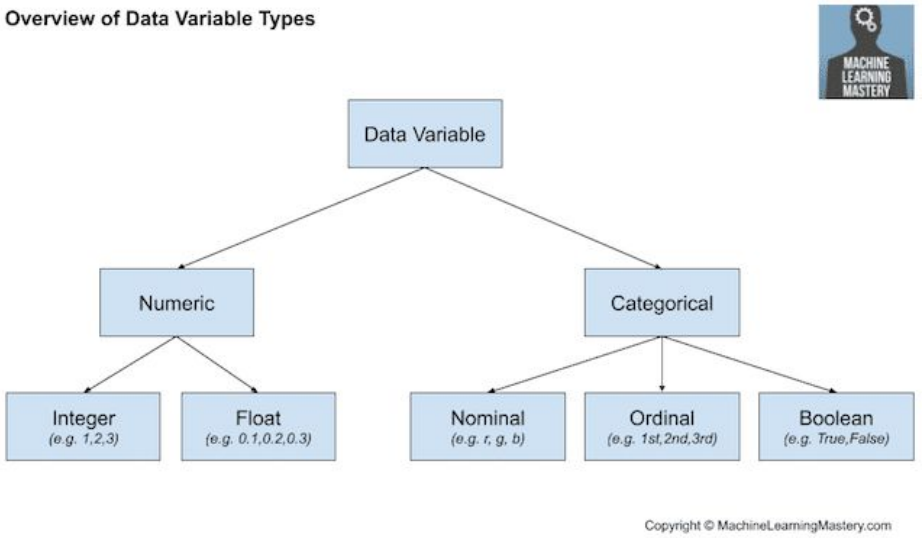
Veri Türleri

Sayısal Veri Türü:

- **Integer**: Kesirli kısmı olmayan tam sayılar
- **Float**: Kayan nokta değerleri

Kategorik Veri Türü:

- **Nominal**: Sıralama içermeyen etiketler
- **Ordinal**: Sıralama sırasına sahip etiketler
- **Boolean**: True ve False değerleri



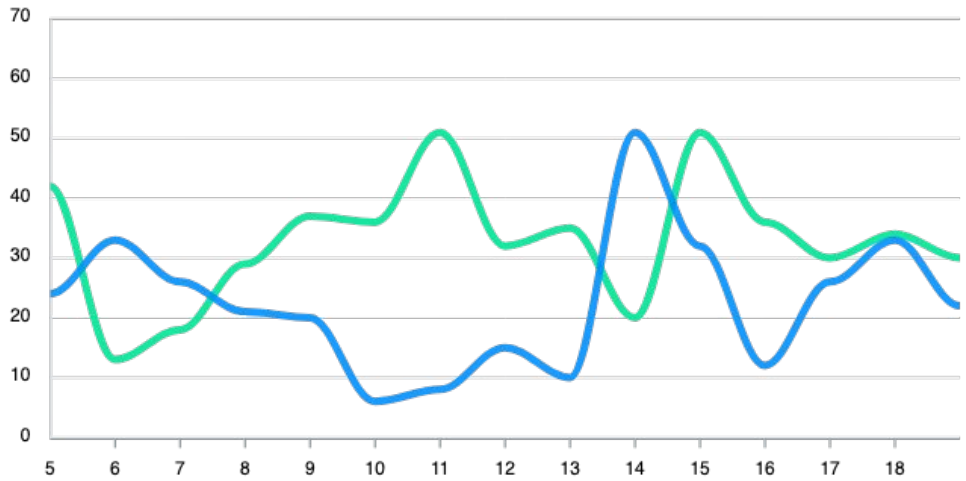
Görselleştirme Tipleri

Line Plot

Çizgi grafiği, temel olarak iki sayısal değer kümesi arasındaki ilişkiyi göstermek için kullanılır. Genellikle, iki bağımlı değişken arasında artan veya azalan bir eğilim göstermek için uygundur.

Örneğin, 24 saatlik bir süre içinde havanın nasıl değiştiğini görmek istiyorsanız, x ekseninin saatlik bilgileri ve y ekseninin derece cinsinden hava durumunu içeren bir çizgi grafiği kullanabilirsiniz.

Verideki trendleri anlamak için bilgileri hızla taramayı kolaylaştırır



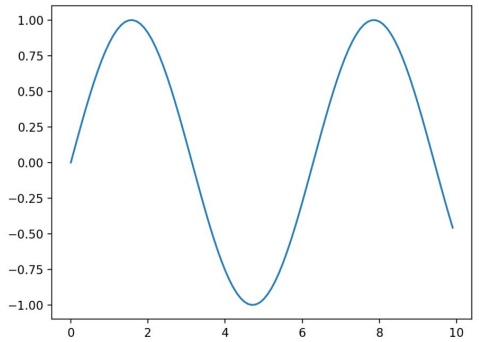
Line Plot Uygulama

Çizgi grafikleri, gözlemler arasında bir sıralamanın olduğu herhangi bir dizi verisinin yanı sıra zaman serisi verilerinin sunulması için kullanışlıdır.

Aşağıdaki örnek, y eksenindeki gözlemler olarak x ekseni ve x ekseninin bir fonksiyonu olarak bir sinüs dalgası olarak 100 kayan nokta değerinden oluşan bir dizi oluşturur. Sonuçlar bir çizgi grafiği olarak çizilir.

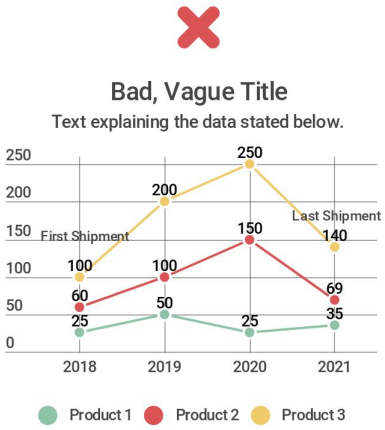
Line Plot

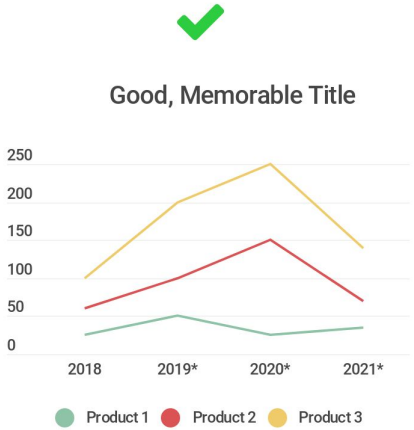
```
1 from numpy import sin
2 from matplotlib import pyplot
3
4 x = [x*0.1 for x in range(100)]
5 y = sin(x)
6
7 pyplot.plot(x, y)
8 pyplot.show()
```



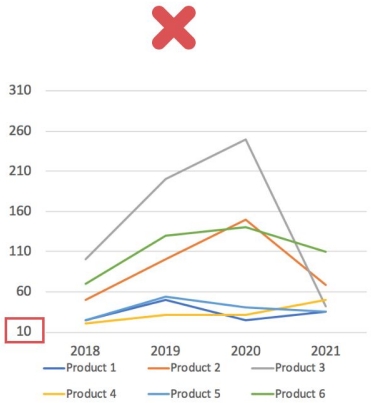
Hatalı Gösterimler

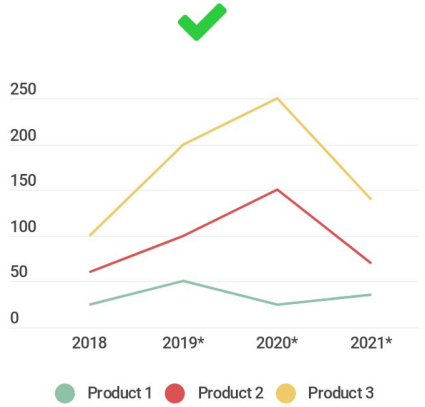
Gereksiz Metin Eklenmesi





Düzgün ve Temi Eksenler





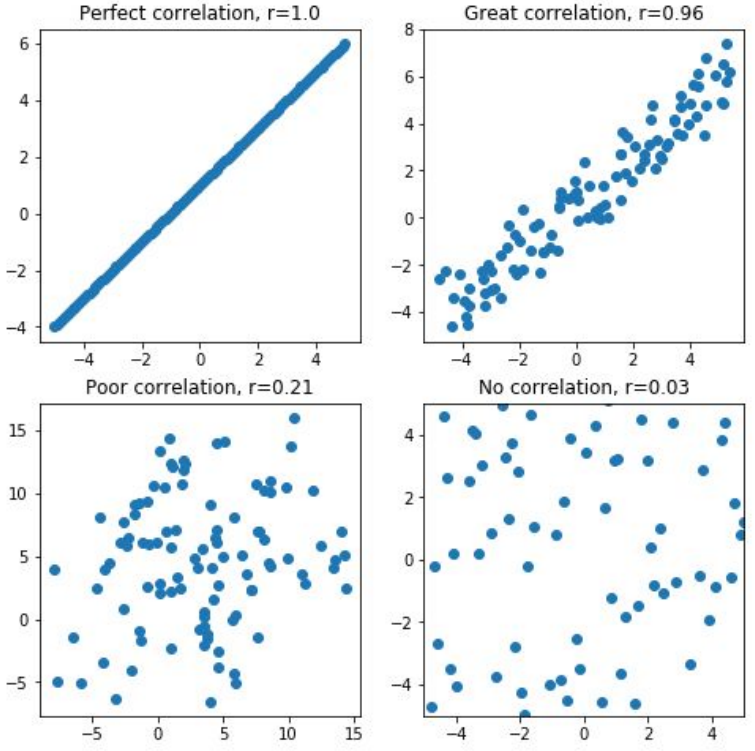
Scatter Plot

Bir Dağılım grafiği, esas olarak iki sayısal grup arasındaki ilişkiyi dağınık noktalar şeklinde çizmek için kullanılır.

Grafikteki her nokta tek bir gözlemi temsil edecek şekilde gösterilir:

- X ekseni örneğin bir özelliğini temsil eder
- Y ekseni, aynı örneğin farklı bir özelliğini temsil eder

Dağılım grafikleri, iki değişken arasındaki ilişkiyi veya korelasyonu göstermek için kullanışlıdır

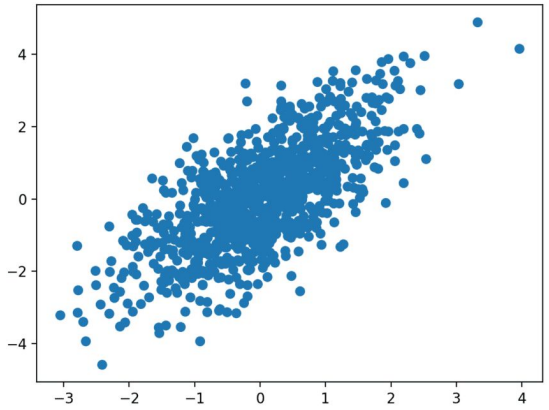


Scatter Plot Uygulama

Aşağıdaki örnek, birbiriyle ilişkili iki veri örneği oluşturur. Birincisi, standart bir dağılımdan alınan rastgele sayıların bir örneğidir. İkincisi, birinci ölçünün değerine ikinci bir rastgele değer ekleyerek birinciye bağlıdır. Sol taraftaki görselde ise bu iki değişkenin korelasyonu ve dağılım grafiği gözükmektedir.

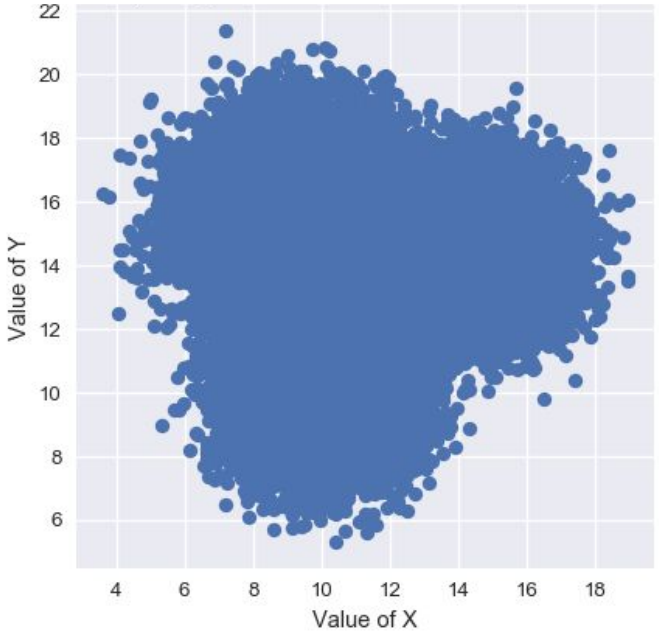
Scatter Plot

```
1 from numpy.random import seed
2 from numpy.random import randn
3 from matplotlib import pyplot
4
5 x = 20 * randn(1000) + 100
6 y = x + (10 * randn(1000) + 50)
7
8 pyplot.scatter(x, y)
9 pyplot.show()
```



Scatter Plot Olası Hatalar

Veri boyutunun yüksek olduğu durumlarda **overplotting** en sık görülen hatadır.

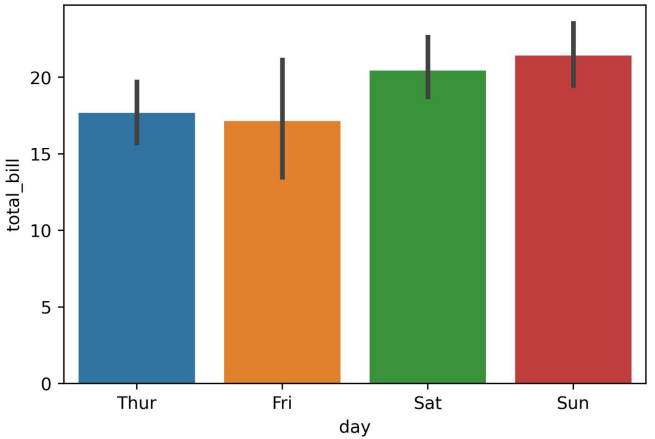


Bar Plot

Çubuk grafiği, toplam, ortalama, medyan vb. gibi bir toplama işleviyle gruplanmış kategorik bir sütundaki benzersiz değerler arasındaki ilişkiyi çizmek için kullanılır.

Kategorik değerler x ekseni olarak iletilir ve karşılık gelen toplu sayısal değerler y ekseninde iletilir.

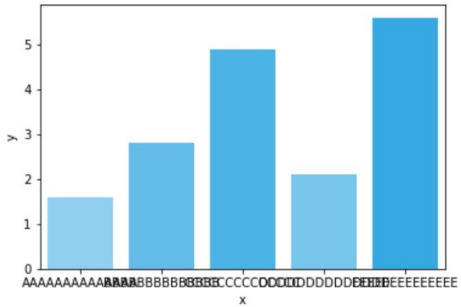
- Farklı kategorideki verileri karşılaştırmak için kullanılır
- Grafiğin bir ekseni karşılaştırılmakta olan belirli kategorileri gösterir ve diğer eksen ölçülen bir değeri temsil eder



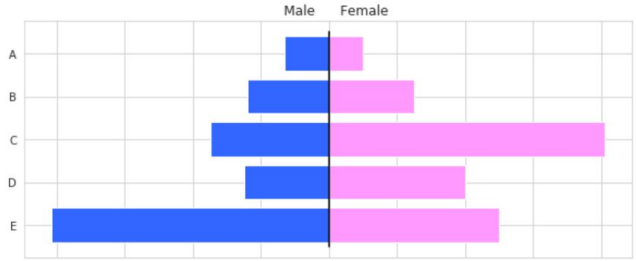
Çubuk grafiğinin histogram ile karıştırılmaması gerekir! Boxplotlar kategorik, histogramlar ise sürekli değerler için kullanılır.

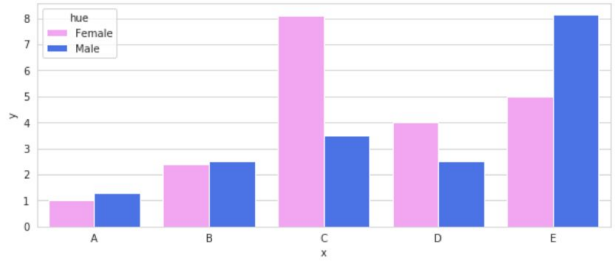
Bar Chart Olası Hatalar

Etiketlerin konumu



Çift yönlü grafikler

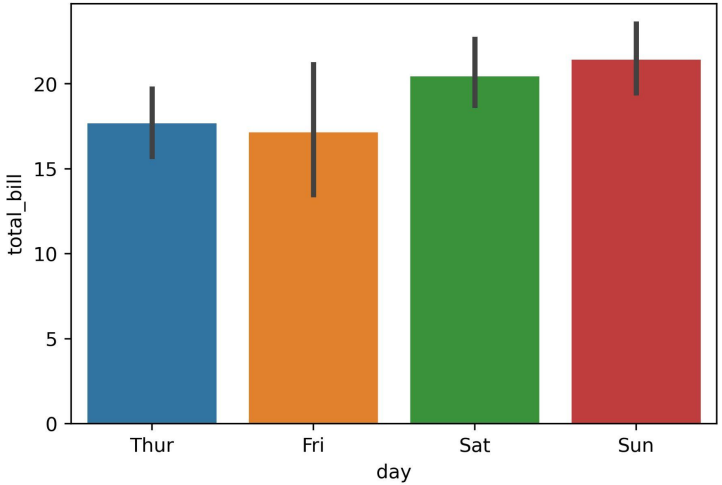




Bar Chart Uygulaması

Kod Örneği

```
tips = sns.load_dataset("tips")
tips.head()
tips.describe().T
sns.barplot(x="day",
            y="total_bill",
            data=tips)
```

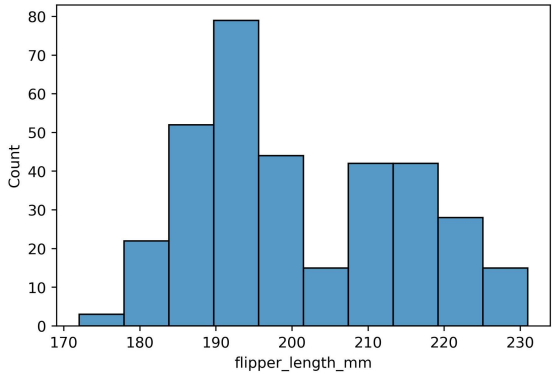


Histogram

Histogramlar temel olarak sayısal bir öğe listesi için veri dağılımını görüntülemek için kullanılır.

Verilerin sürekli bir aralık veya belirli bir süre boyunca dağılımını gösteren bir veri görselleştirme grafiğidir.

- X ekseninde gösterilen sürekli değişken, ayrık aralıklara bölünür ve o ayrık aralıkta sahip olduğunuz veri sayısı, çubuğun yüksekliğini belirler
- Histogramlar, değerlerin nerede yoğunlaştığını, uç noktaların neler olduğunu ve veri kümesinde herhangi bir boşluk veya olağandışı değerler olup olmadığı konusunda bir tahmin verir



$$Bin\ width\ (h) = \frac{3.5 \times \sigma}{\sqrt[3]{n}}$$

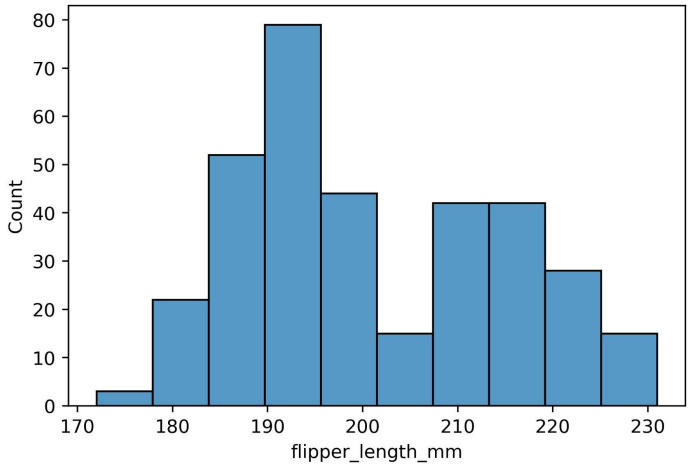
σ = Standard deviation of the data source

n = number of values in the data source

Histogram Uygulaması

Kod Örneği

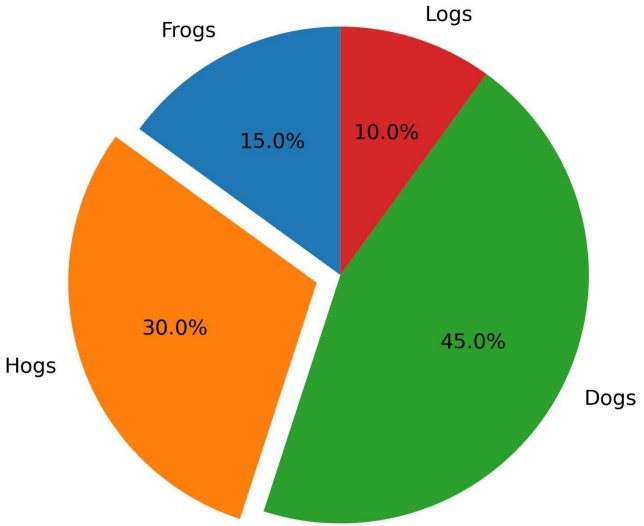
```
penguins = sns.load_dataset("penguins")
penguins.head()
penguins.describe().T
sns.histplot(data=penguins,
             x="flipper_length_mm")
```



Pie Charts

Pasta grafikler, adından da anlaşılacağı gibi, kategorik bir sütundaki değerlerin yüzde dağılımını gösterir.

- Bir bütünün parçaları arasındaki ilişki açık halde görülebilir
- Grafiğin parçaları, her kategorideki bütünün kesri ile orantılıdır

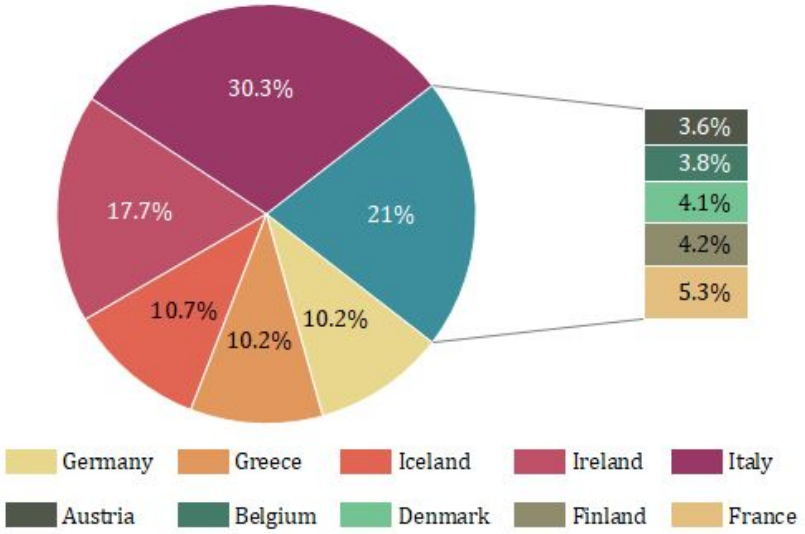


Pie Charts

Sık yapılan hatalar

- Birkaç pasta grafiği karşılaştırılmamalı
- Grafik üzerinde oran gösteriliyorsa, oranlar toplamı 100 olmalıdır

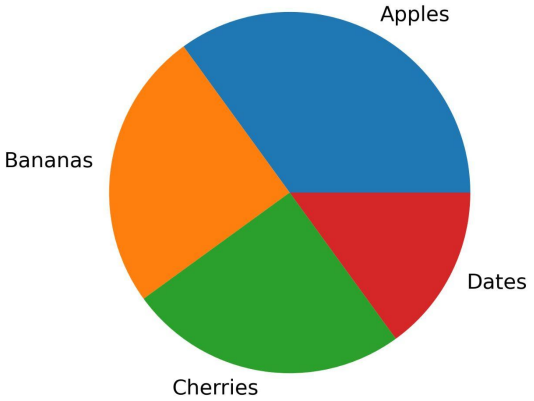
Her dilime açıklama yazılması: Açıklamalar kısa, öz, bir veya birkaç kelime ile açıklanmalıdır



Pie Chart Uygulaması

Kod Örneği

```
import matplotlib.pyplot as plt
import numpy as np

y = np.array([35, 25, 25, 15])
labels = ["Apples", "Bananas",
          "Cherries", "Dates"]

plt.pie(y,labels=labels)
plt.show()
```

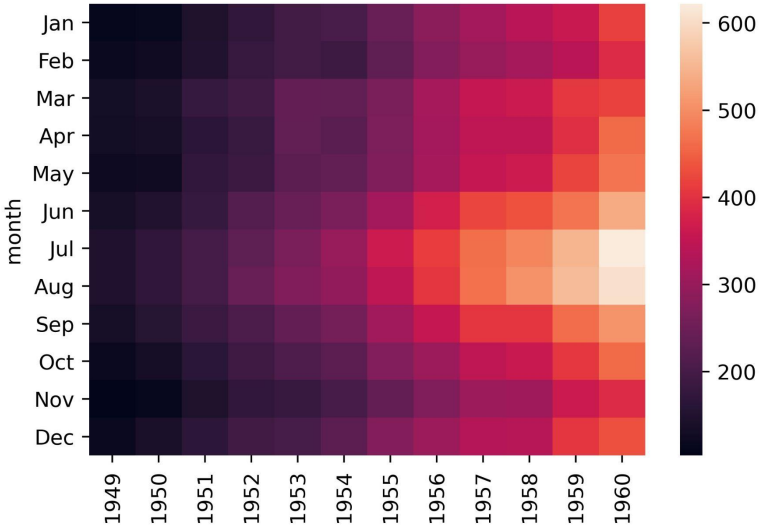


Heatmap

Matrisin değerini görselleştirmek için renkleri kullanarak verilerin grafiksel bir temsilidir

Daha yaygın değerleri veya daha yüksek etkinlikleri temsil etmek için daha parlak renkler kullanılır daha az yaygın veya etkinlik değerlerini temsil etmek için daha koyu renkler tercih edilir.

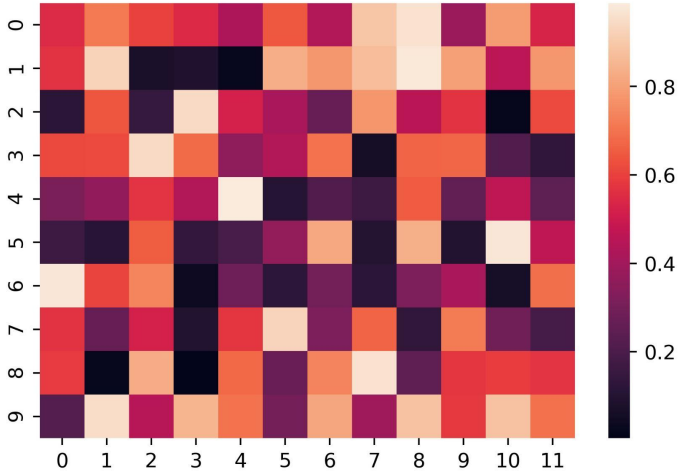
Korelasyon haritalarında veya karmaşıklık matrislerinde sıkça kullanılır.



Heatmap Uygulaması

Kod Örneği

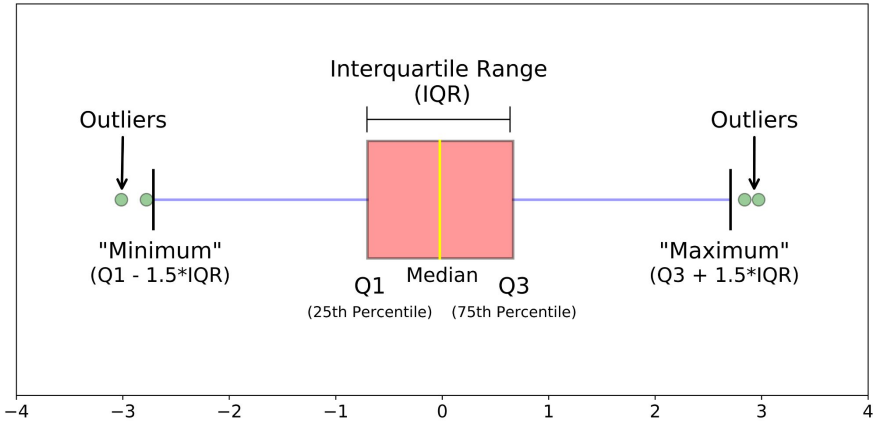
```
import numpy as np

np.random.seed(0)
uniform_data = np.random.rand(10,12)

ax = sns.heatmap(uniform_data)
```



Box Plot

Genellikle gruplar arasında bir veri dağılımını göstermenin görsel bir temsilidir.

- En basit Box plot çizimleri, minimumdan maksimuma tüm varyasyon aralığını, olası varyasyon aralığını ve aykırı değerleri gösterir.
- Box plot beş parçadan oluşur
  - minimum
  - ilk çeyrek
  - medyan (ikinci çeyrek)
  - üçüncü çeyrek
  - maksimum

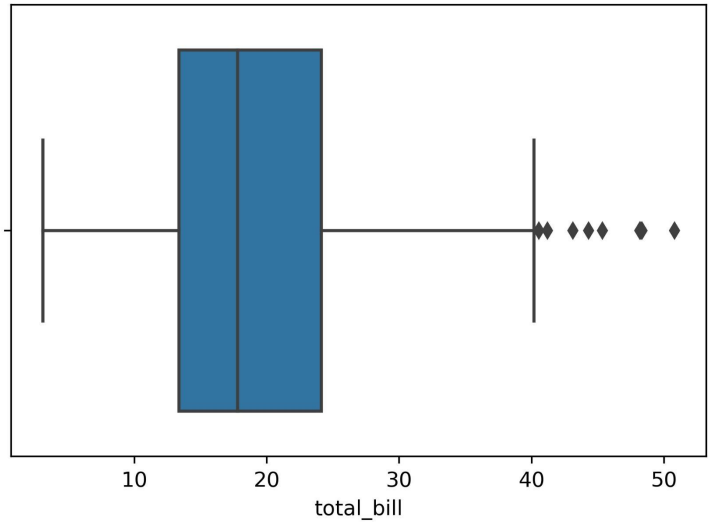


$$\mathrm{IQR} = Q_3 - Q_1$$

Box Plot Uygulaması

Kod Örneği

```
tips = sns.load_dataset("tips")
tips.head()
tips.describe().T
sns.boxplot(x=tips["total_bill"])
```



Baştan Sona Uygulamalı Veri Görselleştirme Projesi