**Data Visualization with Seaborn**

**Seaborn ve Dağılım Grafiklerine Giriş**

Seaborn, Matplotlib tabanlı bir Python veri görselleştirme kütüphanesidir. Çekici ve bilgilendirici istatistiksel grafikler çizmek için üst düzey bir arayüz sağlar. Seaborn'un başlıca avantajları:

Pandas veri DataFrame nesneleri ile iyi çalışır,

Güzel varsayılan stilleri var,

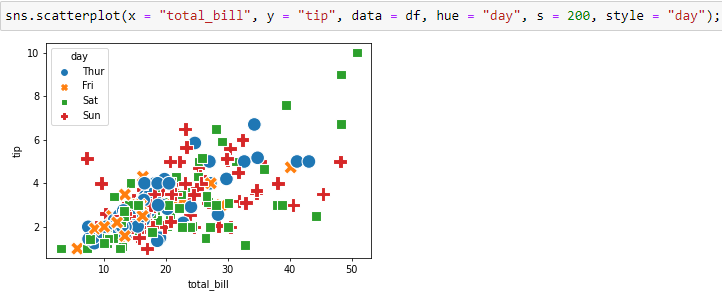
Matplotlib üzerine kurulu olduğu için Matplotlib komutları ile birlikte kullanılabilir,

Kullanımı kolay.

Seaborn plot tiplerini Dağılım Grafikleri, Kategorik Grafikler, Karşılaştırma Grafikleri ve Matris Grafikleri olmak üzere dört ana başlık altında öğreneceksiniz. İstatistikte iki tür veri vardır: sayısal ve kategorik veriler. Yaş, ağırlık, bir şeyin sayısı, ayakkabı numarası, toplam fatura sayısal verilere iyi örneklerdir. Kategorik verilere örnek olarak göz rengi, cinsiyet, haftanın günleri, kan grubu verilebilir.

**Scatter Plot:**

[jupyter notebook#1](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)



**seaborn.scatterplot**(*x=None*, *y=None*, *hue=None*, *data=None*,  *ci=95*, *legend='auto'*)

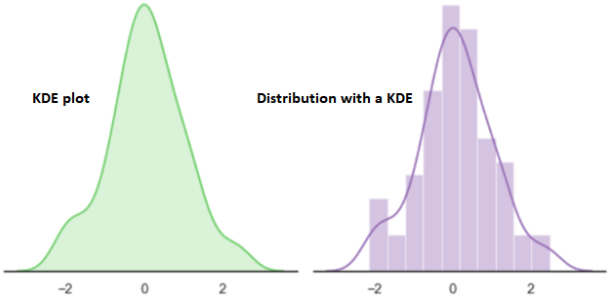
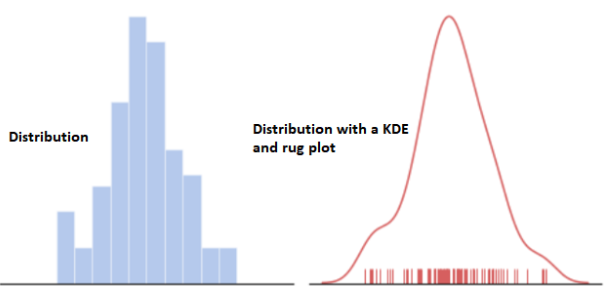
x ve y arasındaki ilişki, ton, boyut ve stil parametreleri kullanılarak verilerin farklı alt kümeleri için gösterilebilir. Bu parametreler, farklı alt kümeleri tanımlamak için hangi görsel anlambilimin kullanıldığını kontrol eder.

Üç semantik türün tümünü kullanarak bağımsız olarak üç boyuta kadar göstermek mümkündür, ancak bu arsa stilini yorumlamak zor olabilir ve genellikle etkisizdir. Gereksiz semantiği kullanmak (yani aynı değişken için hem renk tonu hem de stil), grafikleri daha erişilebilir hale getirmek için yardımcı olabilir.

[Seaborn scatter plot | How to make and style a scatterplot in Python seaborn](https://www.youtube.com/watch?v=4yz4cMXCkuw)

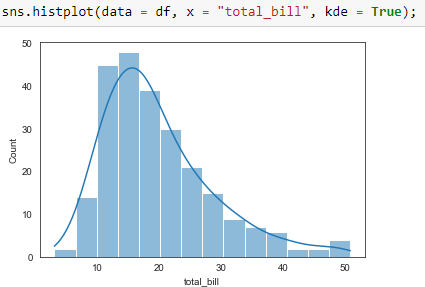
[jupyter notebook#2](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)

**Distribution Plots**



**Histogram:**

[jupyter notebook#3](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)



**seaborn.histplot**(data=None,x=None, y=None, hue=None, bins='auto', kde=False)

Histogram, ayrı kutulara düşen gözlemlerin sayısını sayarak bir veya daha fazla değişkenin dağılımını temsil eden klasik bir görselleştirme aracıdır.

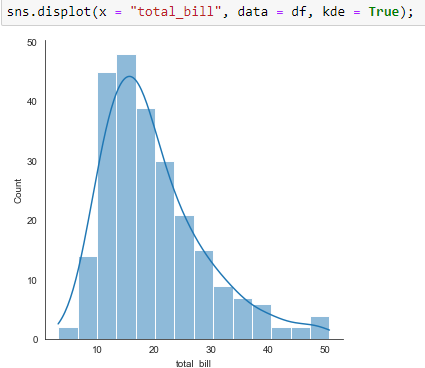
Bu işlev, frekans, yoğunluk veya olasılık kütlesini tahmin etmek için her bir bölmede hesaplanan istatistiği normalleştirebilir ve bir çekirdek yoğunluğu tahmini kullanılarak elde edilen düzgün bir eğri ekleyebilir.

[Seaborn histplot | How to make a Seaborn histogram plot with Python code](https://www.youtube.com/watch?v=Bjz00ygERxY)

[jupyter notebook#4](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)

**Displot:**

[jupyter notebook#5](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)



**seaborn.displot**(data=None, x=None, y=None, hue=None, rug=False)

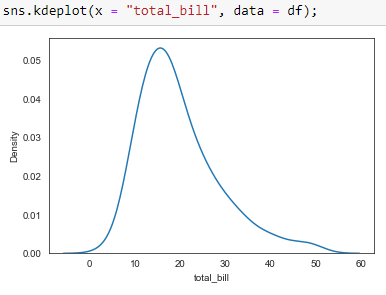
Bu işlev, semantik haritalama ve birden çok alt nokta arasında yüzey oluşturma ile tanımlanan veri alt kümeleri dahil olmak üzere, verilerin tek değişkenli veya iki değişkenli dağılımını görselleştirmek için çeşitli yaklaşımlara erişim sağlar.

[Seaborn displot | What is the displot vs distplot? How to make a Python Seaborn distribution plot](https://www.youtube.com/watch?v=4DA_dgc521o)

[jupyter notebook#6](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)

**Kernel Density Estimere (KDE) plot:**

[jupyter notebook#7](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)

**seaborn.kdeplot**(x=None, y=None, kernel=None, legend=True, hue=None)

kernel density estimate (KDE) grafiği, bir histograma benzer şekilde bir veri kümesindeki gözlemlerin dağılımını görselleştirmek için bir yöntemdir. KDE, bir veya daha fazla boyutta sürekli bir olasılık yoğunluk eğrisi kullanan verileri temsil eder.

Bir histograma göre KDE, özellikle birden fazla dağılım çizerken daha az karmaşık ve daha yorumlanabilir bir çizim üretebilir. Ancak, temeldeki dağılım sınırlıysa veya düzgün değilse, bozulmalar getirme potansiyeline sahiptir. Bir histogram gibi, gösterimin kalitesi de iyi yumuşatma parametrelerinin seçimine bağlıdır.

[What is kernel density estimation? And how to build a KDE plot in Python? | Seaborn KDEplot](https://www.youtube.com/watch?v=DCgPRaIDYXA)

[jupyter notebook#8](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.24.2022)