Outlier verilerin belirlenmesi ve elimine edilmesi

na_values = "?" komutu, bir CSV dosyasını okurken, "?" işaretini NaN (Sayı Olmayan) olarak kabul etmek için kullanılır.

Bu, eksik veya geçersiz verileri temsil etmek için yararlıdır. Pandas, varsayılan olarak ***#N/A, -NaN, -n/a, N/A, NULL*** gibi değerleri de ***NaN*** olarak kabul eder.

Eğer CSV dosyanızda farklı bir değer ***NaN*** olarak kabul edilmek istiyorsanız, na_values parametresine o değeri veya bir liste halinde birden fazla değer verebilirsiniz.

Örneğin, ***na_values = ["not available", "N/A", "0"]*** şeklinde bir komut, ***"not available", "N/A" ve "0"*** değerlerini NaN olarak değiştirir.

---

skipinitialspace = True komutu, bir CSV dosyasını okurken, ayırıcıdan sonra gelen boşlukları atlamak için kullanılır. Bu, ilk sütundaki verilerin doğru şekilde tanınmasını sağlar.

Zaman, Enlem, Boylam

2021-09-12 23:13, 44.63, -63.56

2021-09-14 23:13, 43.78, -62

2021-09-16 23:14, 44.83, -54.6

Bu dosyayı pandas.read_csv fonksiyonuyla okumak isterseniz, ***skipinitialspace = True*** parametresini eklemeniz gerekir. Aksi takdirde, ilk sütundaki zaman değerleri yanlış ayrılacaktır.

---

comment = "\t" komutu, bir CSV dosyasını okurken, "\t" işaretini yorum satırı olarak kabul etmek için kullanılır. Bu, CSV dosyasındaki verilerin başında veya sonunda "\t" işareti olan satırları yoksaymak için yararlıdır.

Pandas, varsayılan olarak "#" işaretini yorum satırı olarak kabul eder. Eğer CSV dosyanızda farklı bir işaret yorum satırı olarak kabul edilmek istiyorsanız, comment parametresine o işareti verebilirsiniz2. Örneğin, comment = "\t" şeklinde bir komut, "\t" işareti ile başlayan veya biten satırları yorum satırı olarak değiştirir.

---

data.isna().sum() komutu, data adlı bir Pandas DataFrame'i olduğunu varsayarsak, DataFrame'deki her sütunda kaç tane eksik değer (NaN) olduğunu hesaplamak için kullanılır.

isna() fonksiyonu, DataFrame'deki her elemanın eksik değer olup olmadığını kontrol eder ve bir True/False değeri döndürür sum() fonksiyonu ise, True/False değerlerini sayısal olarak toplar ve her sütun için bir sonuç verir. Örneğin, şöyle bir DataFrame'imiz olsun:

```
data = pd.DataFrame({'a':[1,2,np.nan], 'b':[np.nan,1,np.nan]})
data
```

```
    a    b
0  1.0  NaN
1  2.0  1.0
2  NaN  NaN
```

Bu DataFrame'de data.isna().sum() komutunu çalıştırdığımızda, şöyle bir sonuç alırız:

```
a    1
b    2
dtype: int64
```

Bu sonuç, a sütununda 1 tane, b sütununda ise 2 tane eksik değer olduğunu gösterir.

Bu komut, data adlı bir Pandas DataFrame'i olduğunu varsayarsak, ***Horsepower*** sütunundaki eksik değerleri (NaN) o sütunun ortalaması ile doldurmak için kullanılır. ***fillna()*** fonksiyonu, DataFrame'deki eksik değerleri belirli bir değer veya yöntem ile ***değiştirirmean()*** fonksiyonu ise, DataFrame'deki sayısal değerlerin ortalamasını hesaplar.

```
data = pd.DataFrame({'Horsepower': [np.nan, 85, np.nan, 88, 94, 90, 76, 75, 87, 86],
'Weight': [25, np.nan, 14, 16, 27, 20, 12, 15, 14, 19]})
data
```

```
   Horsepower  Weight
0         NaN    25.0
1        85.0     NaN
2         NaN    14.0
3        88.0    16.0
4        94.0    27.0
5        90.0    20.0
6        76.0    12.0
7        75.0    15.0
8        87.0    14.0
9        86.0    19.0
```

Bu, DataFrame'de

```
data["Horsepower"]=data["Horsepower"].fillna(data["Horsepower"].mean())
```

komutunu çalıştırdığımızda, şöyle bir sonuç alırız:

data

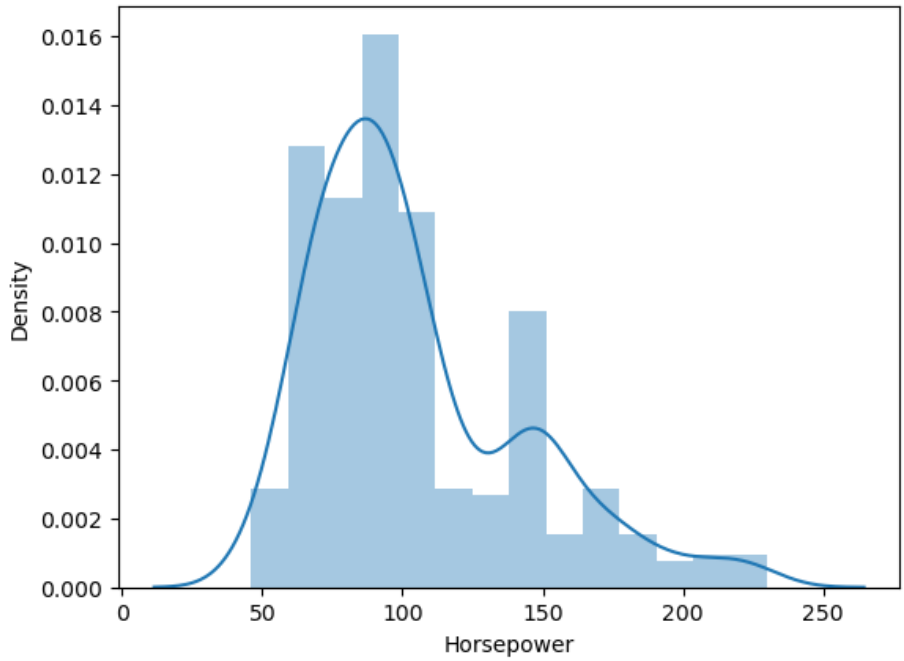
	Horsepower	Weight
0	85.125	25.0
1	85.000	NaN
2	85.125	14.0
3	88.000	16.0
4	94.000	27.0
5	90.000	20.0
6	76.000	12.0
7	75.000	15.0
8	87.000	14.0
9	86.000	19.0

Görüldüğü gibi, ***Horsepower*** sütunundaki eksik değerler, o sütunun ortalaması olan 85.125 ile değiştirildi.

---

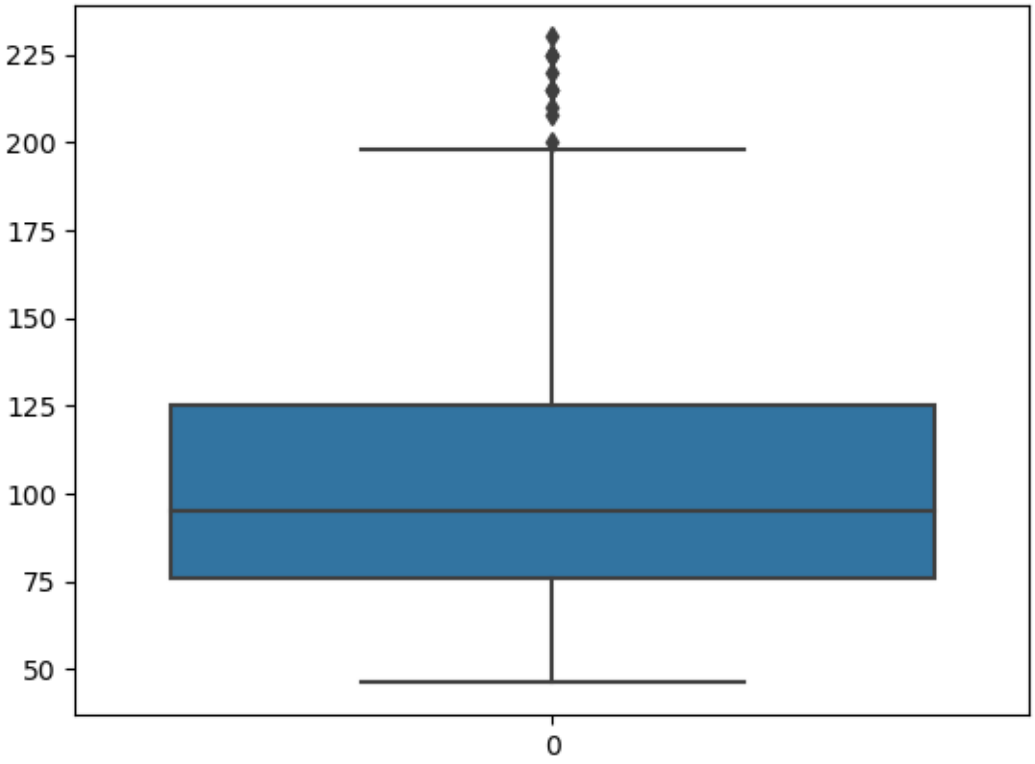
sns.distplot, seaborn kütüphanesinin bir fonksiyonudur. Bu fonksiyon, bir değişkenin dağılımını göstermek için bir histogram ve bir çizgi grafiği çizer. Histogram, verilerin sıklık dağılımını gösterirken, çizgi grafiği, verilerin çekirdek yoğunluk tahmini (KDE) veya başka bir uyum fonksiyonunu gösterir. Bu fonksiyonun nasıl kullanıldığını gösteren bir örnek için buraya bakabilirsiniz.

Ancak, bu fonksiyonun kullanımı kaldırılmıştır ve seaborn v0.14.0 sürümünde kaldırılacaktır. Bu fonksiyonun yerine, ***histplot ()*** ve ***displot ()***, daha modern bir API'ye ve daha fazla özelliğe sahip iki fonksiyon kullanılabilir. Bu fonksiyonları nasıl kullanacağınızı öğrenmek için ***buraya*** bakabilirsiniz.

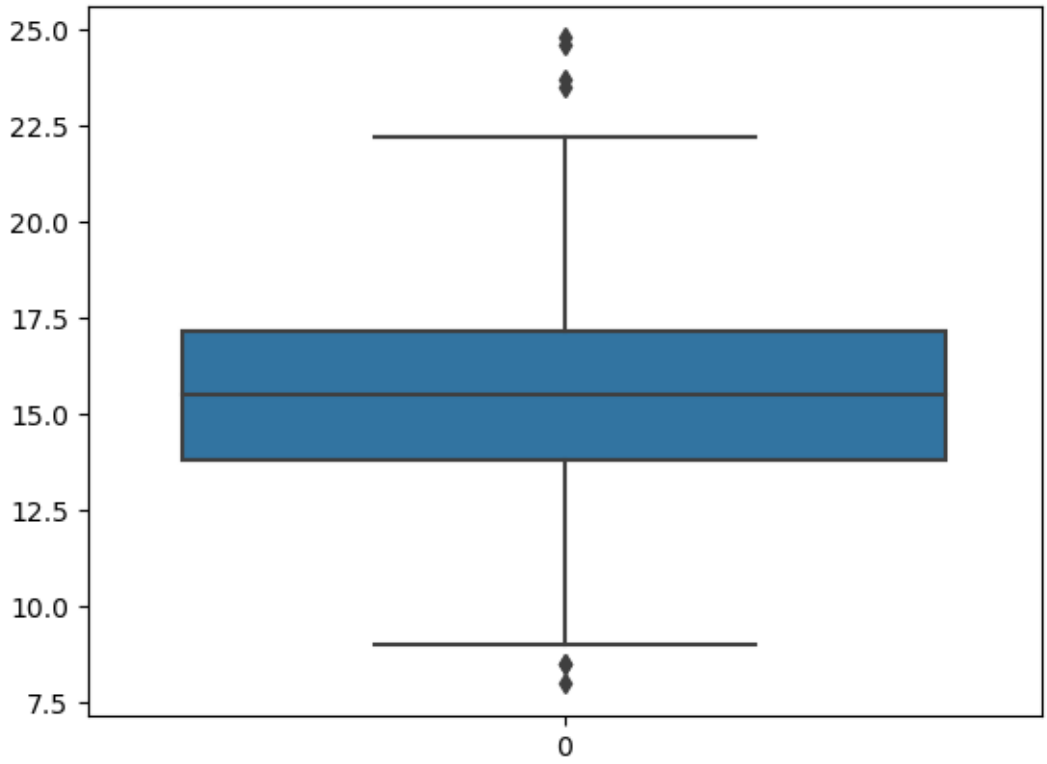


---

sns.boxplot(data = data["Horsepower"])



```
sns.boxplot(data = data["Acceleration"])
```



```
thr = 1.5
# 1.5 denildi ancak çok fazla veri kaybetmek istemiyorsak
# 2 de alabiliriz ????
horsepower_desc = data["Horsepower"].describe()
horsepower_desc
```

```
q3_hp = horsepower_desc[6]
q1_hp = horsepower_desc[4]

IQR_hp = q3_hp - q1_hp
upper_limit_hp = q3_hp + thr * IQR_hp
lower_limit_hp = q1_hp - thr * IQR_hp

filter_hp_bottom = lower_limit_hp < data["xx"]
filter_hp_top = data["xx"] < upper_limit_hp

filter_hp = filter_hp_bottom & filter_hp_top

data = data[filter_hp]
data
```

Bir mum grafikte, quartile 1 ve quartile 3, mumun gövdesinin alt ve üst sınırlarını belirler. Bu değerler, veri kümesinin %25 ve %75 yüzdelik dilimlerine karşılık gelir. Quartile dataları, veri kümesinin dağılımını göstermek için kullanılır. Bir eşit değerine göre quartile dataları atamak için, veri kümesini artan sırada sıralayın ve medyan değerini bulun. Medyan, veri kümesinin ortasındaki değerdir. Medyanı bulduktan sonra, veri kümesini iki alt kümeye bölün: medyandan küçük veya eşit olanlar ve medyandan büyük olanlar. Bu alt kümelerin her birinin medyanını bulun. Bu medyanlar, quartile 1 ve quartile 3 değerleridir. Örneğin, şöyle bir veri kümemiz olsun:

10, 12, 15, 16, 18, 20, 22, 24, 25, 28

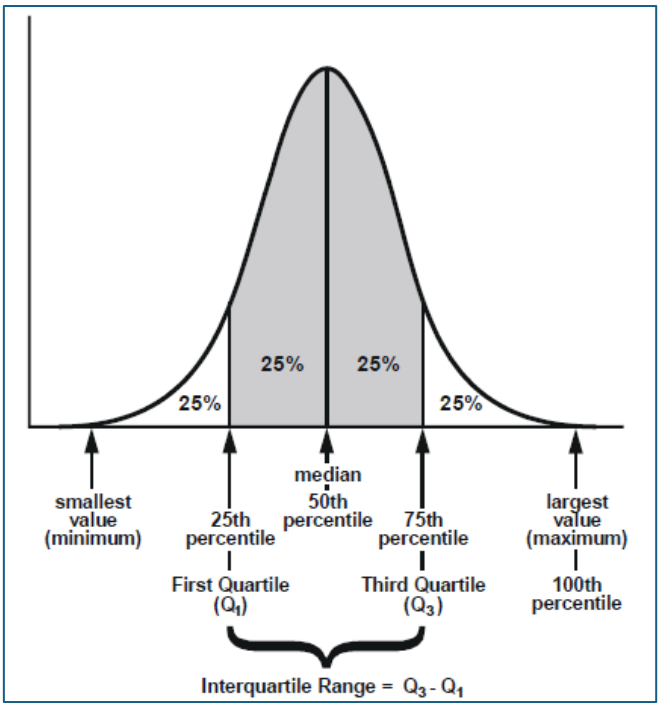
Bu veri kümesinin medyanı, ortadaki iki değerin ortalamasıdır: (18 + 20) / 2 = 19. Bu medyanı kullanarak, veri kümesini iki alt kümeye ayırabiliriz:

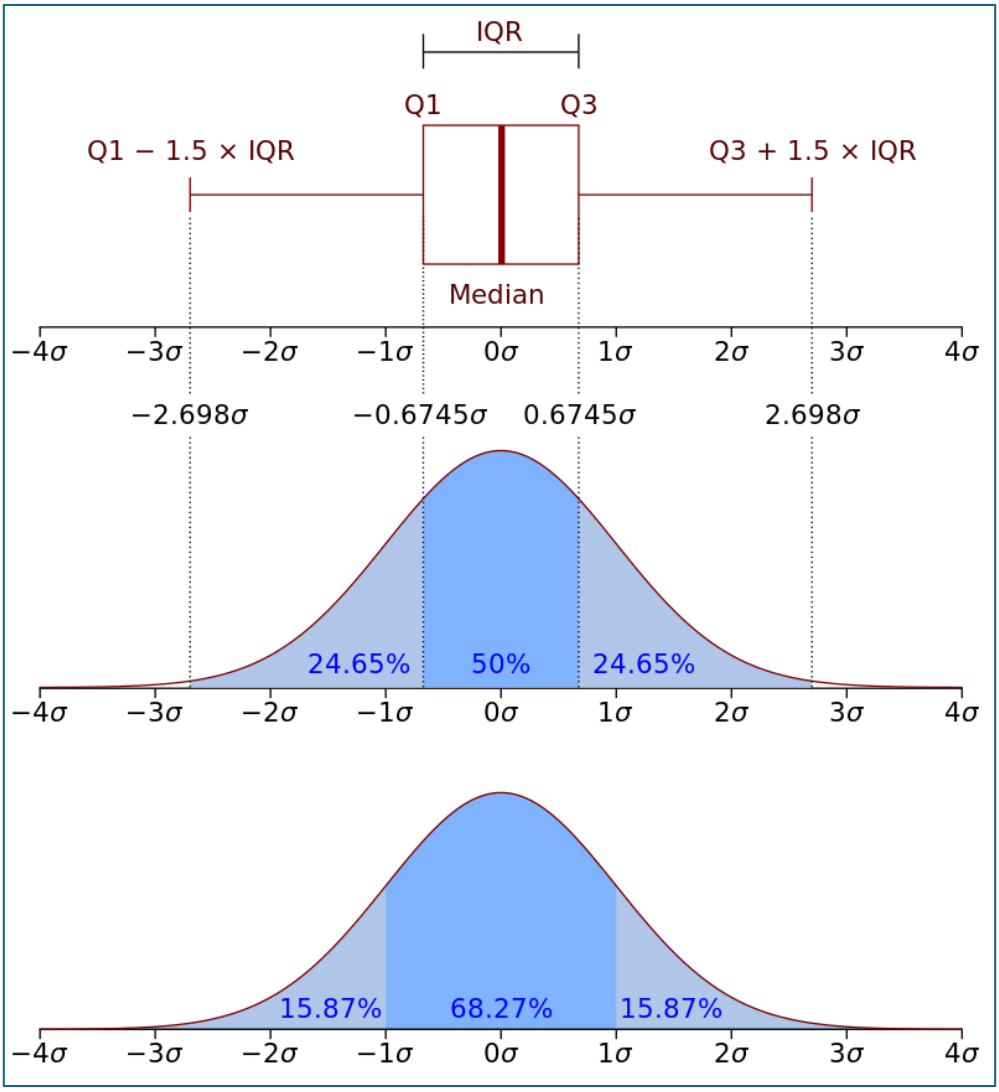
10, 12, 15, 16, 18 | 20, 22, 24, 25, 28

Alt kümelerin medyanları, quartile 1 ve quartile 3 değerleridir:

Q1 = (15 + 16) / 2 = 15.5

Q3 = (24 + 25) / 2 = 24.5


25%
25%
25%
25%
smallest value (minimum)
25th percentile
median 50th percentile
75th percentile
largest value (maximum)
First Quartile (Q_1)
Third Quartile (Q_3)
100th percentile
Interquartile Range = $Q_3 - Q_1$


IQR
Q1
Q3
Q1 − 1.5 × IQR
Q3 + 1.5 × IQR
Median
−4σ −3σ −2σ −1σ 0σ 1σ 2σ 3σ 4σ
−2.698σ
−0.6745σ
0.6745σ
2.698σ
24.65%
50%
24.65%
15.87%
68.27%
15.87%