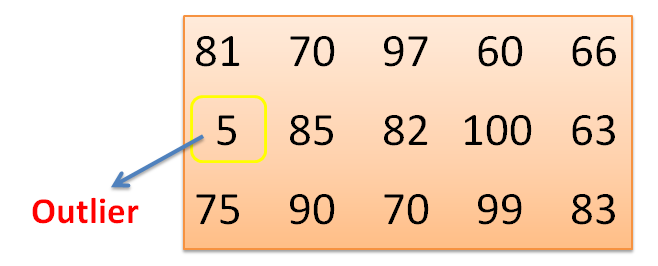
**Cara Menemukan Outlier dengan Python**

*Outlier* atau pencilan adalah data yang nilainya terlalu jauh dengan data lainnya. Keberadaan *outlier* akan membuat analisis terhadap data tersebut menjadi bias atau dengan kata lain tidak mencerminkan fenomena yang sebenarnya.

Contoh *outliers* misalnya nilai mahasiswa dalam satu kelas berkisar antara 60 sampai 100, namun ada satu mahasiswa yang nilainya 5. Nah, nilai 5 ini dianggap sebagai *outliers* karena tentunya akan mempengaruhi statistik data seperti *mean* alias nilai rata-rata kelas. Oleh sebab itu, biasanya data yang mengandung *outliers* akan dihapus.



**Menemukan outlier dengan IQR Score**

Jika data yang kita miliki hanya sedikit seperti contoh di atas, tentunya kita akan dapat dengan mudah menemukan *outlier*-nya. Namun, jika kita bekerja menggunakan dataset yang jumlahnya ratusan bahkan ribuan baris pasti sulit menemukannya.

Nah, Salah satu cara untuk menemukan *outlier* adalah dengan IQR Score.

**IQR** atau *Interquartile Range* adalah selisih dari kuartil ketiga (persentil 75) dengan kuartil pertama (persentil 25). Jika ditulis dalam formula **IQR = Q3 – Q1**.

Di Python, kita dapat menerapkan cara ini dengan beberapa tahap.

**Penerapan IQR Score di Python**

Agar Anda dapat membandingkan hasilnya dengan cara manual, contoh di sini akan menggunakan sedikit data saja.

Sebelumnya kita harus mengimpor Numpy terlebih dahulu.

import numpy as np

Kemudian kita definisikan data yang akan digunakan.

data = [5, 7, 9, 3, 4, -20, 3, 8, 8, 6, 90, 7, 56]

Seperti yang dapat dilihat di atas, ada 13 data yang akan digunakan. Jika kita amati tentunya kita dapat mengetahui dari rentetan bilangan tersebut yang termasuk *outlier* adalah -20, 90, dan 56, bukan?!

Sekarang bagaimana kode Python-nya untuk mendapatkan bilangan-bilangan tersebut?

**Cari Q1 dan Q3, kemudian hitung IQR**

Pertama kita harus mencari kuartil 1 dan kuartil 3 dari data tersebut. Kita dapat menggunakan [quantile()](https://numpy.org/doc/stable/reference/generated/numpy.quantile.html) dari Numpy seperti di bawah ini.

*# Hitung nilai Q1 dan Q3*  
Q1 = np.quantile(data, .25)  
Q3 = np.quantile(data, .75)

Selanjutnya kita cari nilai IQR-nya.

*# Hitung nilai IQR*  
IQR = Q3 - Q1

Sekarang mari kita *print* hasilnya.

print('Kuartil 1 = ', Q1)  
print('Kuartil 3 = ', Q3)  
print('IQR = ', IQR)

**Output:**

    Kuartil 1 =  4.0  
    Kuartil 3 =  8.0  
    IQR =  4.0

**Hitung IQR minimum dan maksimum**

Selanjutnya, kalikan IQR dengan **1.5** (konstanta untuk menemukan *outliers*) untuk mendapatkan nilai IQR minimum dan maksimum.

Untuk IQR minimum, kurangi Q1 dengan **1.5 x IQR**, sedangkan untuk IQR maksimum, tambahkan Q3 dengan **1.5 x IQR**

min\_IQR = Q1 - 1.5 \* IQR  
max\_IQR = Q3 + 1.5 \* IQR  
  
print('IQR minimum = ', min\_IQR)  
print('IQR maksimum = ', max\_IQR)

**Output:**

    IQR minimum =  -2.0  
    IQR maksimum =  14.0

**Cari nilai minimun dan maksimum dari data**

Kemudian cari nilai minimum dan maksimum dari data.

nilai\_min = np.min(data)  
nilai\_max = np.max(data)  
  
print('Nilai minimum data = ', nilai\_min)  
print('Nilai maksimum data = ', nilai\_max)

**Output:**

    Nilai minimum data =  -20  
    Nilai maksimum data =  90

**Buat kondisi untuk mendefinisikan outlier**

Selanjutnya, buat kondisi di mana jika nilai minimum data lebih kecil dari IQR minimum, berarti ada *low outlier*, dan jika nilai maksimum data lebih besar dari IQR maksimum, berarti ada *high outlier.*

if (nilai\_min < min\_IQR):  
    print('Ditemukan low outlier!')  
else:  
    print('Tidak ditemukan low outlier!')    
  
if (nilai\_max > max\_IQR):  
    print('Ditemukan high outlier!')  
else:  
    print('Tidak ditemukan high outlier!')

**Output:**

    Ditemukan low outlier!  
    Ditemukan high outlier!

Dapat kita lihat bahwa ada *low outlier* atau data yang nilainya jauh di bawah *range* nilai normal, dan juga *high* *outlier* atau data yang nilainya jauh di atas *range* nilai normal.

Sekarang mari kita tampilkan data yang termasuk *outlier*.

low\_out = []  
high\_out = []  
  
for i in data:  
    if (i < min\_IQR):  
        low\_out.append(i)  
    if (i > max\_IQR):  
        high\_out.append(i)  
  
print('Low outlier : ', low\_out)  
print('High outlier : ', high\_out)

**Output:**

    Low outlier :  [-20]  
    High outlier :  [90, 56]

Dapat kita lihat bahwa dengan menggunakan IQR Score, juga didapatkan 3 *outlier* yaitu **-20**, **90**, dan **56**. Itulah cara penerapan IQR untuk menemukan outlier menggunakan Python.