

Beräkning av medelvärde och median

Medelvärdet är det vanligaste sättet att sammanfatta en uppsättning data. Det är vad vi helt enkelt kan kalla genomsnittet. Att beräkna medelvärdet är inte speciellt svårt. Med grafräknaren så matar du säkert in dina data i statistikeditorn och sedan beräknar du medelvärde och median i grundfönstret.

Det finns andra typer av genomsnitt också. Låt oss ta en nu direkt på sak beräkna medelvärdet av en uppsättning data. Det handlar om vikter hos alla nyfödda barn under en vecka på en förlossningsklinik. Att beräkna medelvärdet är ju bara att addera alla vikter och sedan dividera med antalet nyfödda. Inte särskilt svårt men här handlar det ju om att skriva ett snyggt Pythonprogram.

Här är alla data och i tidsordning:

4411,2769, 2892, 3287, 1969, 3980, 4101, 3129, 3459, 2919, 3068, 3821, 4139, 3837, 3782, 4025, 4220, 4538, 2788, 4022

Vi går nu igenom programmet i steg från början:

1. Först så definierar vi en funktion **beräkna_mvärde()** som tar emot alla argumentvärden, som finns i en lista.
2. Sedan på rad 2 och 3 så använder vi funktionerna **sum** och **len** för att summera alla värden och beräkna antalet poster i listan.
3. Här gör vi en beräkning av mvärdet genom att dividera summan (**s**) med antalet värden (**n**)
4. Nu skapar vi en lista med vikter, där vi matar in alla vikter och sedan anropar **beräkna_mvärde()**-funktionen och lägger listan som argument i funktionen
5. I printsatsen så skriver vi ut medelvärdet som beräknades tidigare. Observera att vi med **{0}** och **{1}** kan styra hur utskriften blir. Funktionen **round** är en avrundningsfunktion. Vi vill alltså ha 0 decimaler vid utskrift.

Beräkna medianen

Medianen för en uppsättning data är en annan typ av genomsnitt. För att hitta medianen sorterar vi data i stigande ordning. Om längden på listan med data är udda är data i mitten av listan medianen. Om längden på listan är ett jämnt antal får vi medianen genom att ta medelvärdet av de två data i mitten. Låt oss hitta medianen för den tidigare listan över födelsevikter:

L1	L2	L3	L4	L5	2
4411					
2769					
2892					
3287					
1969					
3980					
4101					
3129					
3459					
2919					
3068					
3821					
4139					
3837					
3782					
4025					
4220					
4538					
2788					
4022					

NORMAL FLYT DEC REELL RAD MP	
medel(L1)	3557.8
median(L1)	3801.5

```

EDITOR: MEAN
PROGRAM LINE 0001
def beräkna_mvärde(värden):
    s = sum(värden)
    n = len(värden)
    # Beräkna medelvärdet
    mvärde = s/n
    return mvärde
vikter = [4411,2769,2892,3287,1969,3980,4101,3129,3459,2919,3068,3821,4139,3837,3782,4025,4220,4538,2788,4022]
mvärde = beräkna_mvärde(vikter)
print('Genomsnittet för de senaste {0} bidragen är {1}'.format(N, round(mvärde,0)))
    
```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running MEAN
>>> from MEAN import *
Genomsnittet för de senaste 20
vikterna är 3558.0
>>> |
    
```

Efter sorteringen från minsta till största blir listan så här:

1969, 2769, 2788, 2892, 2919, 3068, 3129, 3287, 3459, 3782, 3821, 3837, 3980, 4022, 4025, 4101, 4139, 4220, 4411, 4538

Vi har ett jämnt antal poster i listan (20), så för att få medianen måste vi ta medelvärdet av de två mellersta posterna (markerade med gult), i detta fall är talen i mitten den tionde och elfte dvs 3782 och 3821 och medelvärdet av dessa två tal är $(3782 + 3821)/2 = 3801,5$.

Innan vi skriver ett program för att hitta medianen för en lista med tal så kan vi oss tänka det allmänna fallet genom att använda n som antalet tal i listan. Om längden på listan (n) är udda, är det mellersta talet $(n + 1)/2$.

Om n är jämnt är de två mellersta talen i den sorterade listan $n/2$ och $n/2 + 1$. I vårt fall med $n = 20$ så blir det ju tal 10 och 11.

För att skriva ett program som beräknar medianen så måste vi ha en funktion som kan sortera. Det finns nu en sådan i Python. Den heter `sorted()`. Se skärmbilden till höger.

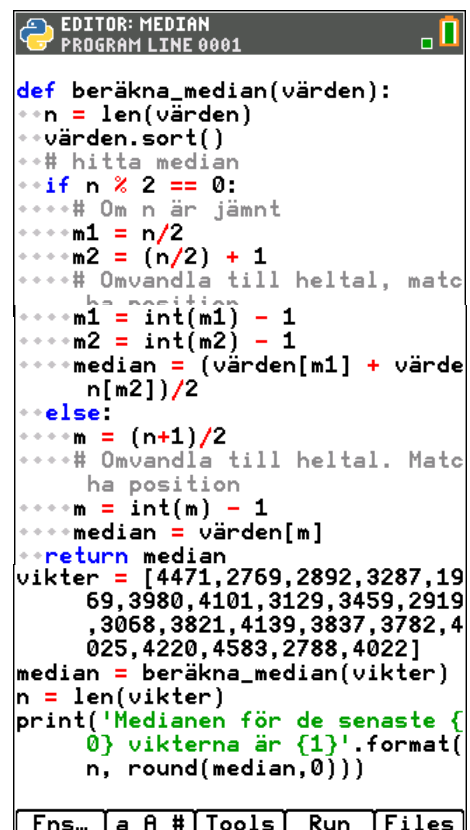
Nu börjar vi bli klara med att sätta ihop programmet för beräkning av medianen. Den övergripande strukturen i programmet liknar den i programmet som beräknar den medelvärdet. **beräkna_median()**-funktionen tar emot en lista tal och returnerar medianen.

1. Först så beräknar vi längden, antalet poster, och skapar en etikett, n , som refererar till listan.
2. Därefter så sorteras denna lista i stigande ordning med funktionen `sorted()`.
3. Sedan kontrollerar man om n är jämnt. Om det är så hittar vi de mellersta posterna $m1$ och $m2$ som är talen vid lägena $n/2$ och $n/2 + 1$ i den sorterade listan.
4. I nästa två satser så justeras $m1$ and $m2$ på två sätt. Först med funktionen `int()` som omvandlar $m1$ and $m2$ till heltal. Det beror på att resultat där man använt divisionsoperatören `/` returneras som ett flyttal trots att resultatet är ett flyttal. Prova att t ex dividera 12 med 3. Vi får resultatet 4.0. Vi måste alltså använda `int`-funktionen.
5. Vi subtraherar också 1 från både $m1$ och $m2$ eftersom positioner i en lista *börjar med 0* i Python.
6. Sedan beräknar vi medianen genom att ta medelvärdet av de två talen i mitten.
7. Efter `else`-satsen så genomförs motsvarande beräkningar om antalet tal i listan är udda. Allt blir lite enklare eftersom vi bara behöver göra beräkningar med ett tal.
8. Slutligen så beräknas medianen och returnerar resultatet.



```

PYTHON SHELL
>>> minlista=[3,2,1,5,1,0]
>>> sorted(minlista)
[0, 1, 1, 2, 3, 5]
>>> |
  
```



```

EDITOR: MEDIAN
PROGRAM LINE 0001

def beräkna_median(värden):
    n = len(värden)
    värden.sort()
    # hitta median
    if n % 2 == 0:
        # Om n är jämnt
        m1 = n/2
        m2 = (n/2) + 1
        # Omvandla till heltal, matcha position
        m1 = int(m1) - 1
        m2 = int(m2) - 1
        median = (värden[m1] + värden[m2])/2
    else:
        m = (n+1)/2
        # Omvandla till heltal. Matcha position
        m = int(m) - 1
        median = värden[m]
    return median

vikter = [4471,2769,2892,3287,1969,3980,4101,3129,3459,2919,3068,3821,4139,3837,3782,4025,4220,4583,2788,4022]
median = beräkna_median(vikter)
n = len(vikter)
print('Medianen för de senaste {0} vikterna är {1}'.format(n, round(median,0)))
  
```

Ändra nu i listan över värden så att vi har ett udda antal värden och kör sedan programmet. Kontrollera resultatet genom att göra beräkningen utanför Pythonappen.

Bilden nedan visar fördelningen av vikterna i ett punktdiagram. Vi ser att spridningen är ganska stor. De vertikala linjerna visar medelvärdet och medianen.

