

# Data V - inleveropdracht 1

Boris van Boxtel, Brechtje Poppen,  
Floris Oostenbrug, Lotte Gritter

9 December 2022 - Week 49

- (a.) Om de gemiddelde valtijd te berekenen gebruiken we python. Ook berekenen we de standaarddeviatie door middel van python. Dit geeft een waarde voor de gemiddelde valtijd ( $\bar{t}$ ) van 72.9 in honderdste secondes en voor de standaarddeviatie ( $\sigma_t$ ) van 7.037627805461519 in honderdste secondes.

We gebruiken nu de tien procent regel om de standaarddeviatie correct af te ronden. Hiertoe ronden we de onzekerheid eerst af op twee decimalen, dus  $\sigma_t = 7.04/100$  s. Wanneer we dit nu op één decimaal afronden krijgen we  $7.0/100$  s. Volgens de tien procent regel mag je alleen op één decimaal afronden als hierdoor de afrondingsfout kleiner is dan tien procent van de onzekerheid in twee decimalen omdat er dan geen significant verlies van informatie is. Tien procent van de onzekerheid in twee decimalen is  $0.704/100$  s en de afrondingsfout is  $0.04/100$  s, wat veel kleiner is dan  $0.704/100$  s. De onzekerheid is dus  $7.0/100$  s. Dit betekent dat de gemiddelde valtijd afgerond moet worden op  $73/100$  s.

De waarde van  $\bar{t}$  is dus  $(73 \pm 7.0)/100$  s.

- (b.) We berekenen nu  $\bar{t}_1, \dots, \bar{t}_{10}$  voor de 4 metingen in elk van de 10 kolommen door middel van python. Dit geeft de waarden:  $\bar{t}_1 = 74.25/100$  s,  $\bar{t}_2 = 67.75/100$  s,  $\bar{t}_3 = 72.5/100$  s,  $\bar{t}_4 = 77.25/100$  s,  $\bar{t}_5 = 66.25/100$  s,  $\bar{t}_6 = 74.75/100$  s,  $\bar{t}_7 = 75.75/100$  s,  $\bar{t}_8 = 76./100$  s,  $\bar{t}_9 = 71.75/100$  s en  $\bar{t}_{10} = 72.75/100$  s.
- (c.) We hebben in python het gemiddelde ( $\bar{t}_{deel}$ ) en de bijbehorende standaarddeviatie ( $\sigma_{deel}$ ) berekend. Dit gaf de waarden  $\bar{t}_{deel} = 72.9$  in honderdste secondes en  $\sigma_{deel} = 3.561366778827103$  in honderdste secondes.

We gebruiken nu de van procent regel om de standaarddeviatie goed af te ronden. We ronden de standaarddeviatie eerst af op  $3.56/100$  s. Wanneer we dit op  $3.6/100$  s afronden is de afrondfout gelijk aan  $0.04/100$  s. Tien procent van  $3.56/100$  s is gelijk aan  $0.356/100$  s. De afrondingsfout van  $0.04/100$  s is veel kleiner dan  $0.356/100$  s dus we ronden de standaarddeviatie af op  $3.56/100$  s. Dit betekent dat  $\bar{t}_{deel}$  afgerond wordt naar  $73/100$  s.

Het is wel te verwachten dat het gemiddelde van de valtijden berekend in (a.) gelijk is aan het zojuist berekende  $\bar{t}_{deel}$ . Dit is ook het geval. Dit is te verwachten omdat we bij opdracht (a.) de som van  $t_1, \dots, t_{40}$  hebben gedeeld door 40. We hebben dus de berekening  $\frac{1}{40} \sum_{n=1}^{40} t_i = \bar{t}$  gedaan om  $\bar{t}$  te krijgen.

In opdracht (b.) hebben we toen de berekeningen  $\frac{1}{4}\sum_{n=1}^4 t_i = \bar{t}_1$ ,  $\frac{1}{4}\sum_{n=5}^8 t_i = \bar{t}_2$ ,  
 $\dots$ ,  $\frac{1}{4}\sum_{n=37}^{40} t_i = \bar{t}_{10}$  gedaan om  $\bar{t}_1, \dots, \bar{t}_{10}$  te krijgen.

Toen hebben we in opdracht (c.) de berekening  $\frac{1}{10}\sum_{n=1}^{10} \bar{t}_i = \bar{t}_{deel}$  gedaan om  $\bar{t}_{deel}$  te krijgen.

De berekening die in opdracht (c.) is gedaan is te schrijven als  $\frac{1}{10} \cdot (\frac{1}{4}\sum_{n=1}^4 t_i + \frac{1}{4}\sum_{n=5}^8 t_i + \dots + \frac{1}{4}\sum_{n=37}^{40} t_i) = \frac{1}{40} \cdot (\sum_{n=1}^4 t_i + \sum_{n=5}^8 t_i + \dots + \sum_{n=37}^{40} t_i) = \frac{1}{40}\sum_{n=1}^{40} t_i$ .  
De berekening die we bij (c.) hebben gedaan is dus gelijk aan de berekening die we bij (a.) deden en dus is het te verwachten dat de uitkomst gelijk is.

Het is echter niet te verwachten dat de uitkomst voor de standaarddeviatie bij (a.) en (c.) gelijk zou zijn. Dit is ook niet het geval.

Wanneer namelijk het gemiddelde van vier valtijden genomen wordt ligt de uitkomst dichterbij de werkelijke valtijd dan elk van de vier losse metingen. De waarden waarvan het gemiddelde werd genomen in (c.) lagen dus allemaal al dichterbij de werkelijke valtijd dan de losse metingen bij (a.). Hoe dichterbij de waarden waarover de standaarddeviatie wordt berekend bij het werkelijke gemiddelde liggen, hoe kleiner de standaarddeviatie. De standaarddeviatie bij (c.) was inderdaad kleiner dan bij (a.).

- (d.) Er geldt:  $t_5 = (70 \pm 7.0)/100$  s en  $t_8 = (68 \pm 7.0)/100$  s. Wanneer twee metingen consistent zijn met elkaar kunnen ze door de onzekerheid dezelfde waarde hebben. Wanneer twee metingen strijdig zijn kunnen ze niet dezelfde waarde hebben, zelfs niet wanneer er rekening gehouden wordt met de onzekerheid. Aangezien de waarde van  $t_5$  tussen 63.0/100 s en 77.0/100 s ligt en die van  $t_8$  tussen 61.0/100 s en 75.0/100 s kunnen ze dezelfde waarde hebben. De resultaten  $t_5$  en  $t_8$  zijn dus consistent. (niet af)

code/DataV\_inlever1.py

---

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

#a
#onderstaande numpy array bevat de gemeten valtijden t1 tot
#t40

valtijden = np.array([63, 58, 74, 78, 70, 74, 75, 82, 68,
                      69, 76, 62, 72, 88, 65, 81, 79, 77, 66, 76, 86, 72, 79,
                      77, 60, 70, 65, 69, 73, 77, 72, 79, 65, 66, 70, 74, 84,
                      76, 80, 69])

#Nu berekenen we de gemiddelde valtijd, tgem
tgem = np.mean(valtijden)
print(
    'De gemiddelde valtijd zoals berekend door python is ',
    tgem,
    ' in honderste secondes.'
)

#Nu berekenen we de standaarddeviatie:
```

```

tsd = np.std( valtijden , ddof=1)
print(
    'De standaarddeviatie is zoals berekend door python is '
    ,
    tsd ,
    'in honderste secondes . '
    )

#b
#We herschikken onze numpy array zodat deze wordt opgedeeld
in tien kolommen.
valtijdenk = valtijden.reshape(4,10)
#We maken hier een numpy array van zodat we straks de
standaarddeviatie uit
#kunnen rekenen.
valtijdenknp = np.array(valtijdenk)

#Nu berekenen we de gemiddeldes en standaarddeviatie van
deze arrays.

deeltgem = np.mean(valtijdenknp , axis=0)

#c

#We bereken de gemiddelde valtijd en de standaarddeviatie
van de 10 deel
#experimenten:
tdeelgem = np.mean(deeltgem)

tdeelstd = np.std(deeltgem , ddof=1)

```

---