

Dit document geeft een Python code voor het simuleren van het lottoprobleem uit paragraaf 5.2.3 van het boek.

```
import numpy as np
import numpy.random as rnd

def simulatierun(rng):
    '''
    Deze functie simuleert een trekking uit de 6/45 lotto.
    rng = instantie van random number generator.
    De functie retourneert 1 als er twee opeenvolgende getallen
    tussen zijn, anders 0.
    '''
    Y = 0 # uitvoer wordt 1 als succes
    lotto = rng.choice(45, size=6, replace=False)
        # random 6 verschillende getallen uit 45
    rij = np.sort(lotto) # gesorteerd op grootte

    for i in range(5):
        if rij[i+1] == rij[i]+1:
            Y = 1
            break
    return Y

def simulatie(nrun):
    '''
    Deze functie simuleert nrun keer trekkingen uit 6/45 lotto,
    en houdt dan bij in welke trekking er twee opeenvolgende getallen
    zijn. Vervolgens wordt de output statistiek uitgevoerd voor
    het schatten van de bijbehorende kans.
    '''
    rng = rnd.default_rng() # instantie van RNG
    Y = np.zeros(nrun) # aanmaak vector Y

    for i in range(nrun):
        Y[i] = simulatierun(rng)

    est = np.mean(Y) # geschatte kans
    S2 = np.var(Y, ddof=1) # steekproefvariantie
    SE = np.sqrt(S2 / nrun) # standaardfout
    print('geschatte kans: %.4f' % est)
    print('95%% BTI : (%.4f,%.4f)' % (est-1.96*SE, est+1.96*SE))

def main():
    nrun = 1000000
    simulatie(nrun)
```

```
if __name__ == '__main__':  
    main()
```

Uitvoer

```
geschatte kans: 0.5292  
95% BTI        : (0.5282,0.5302)
```

Opmerking: In bovenstaande code worden zes willekeurige getallen uit $1, \dots, 45$ getrokken door aanroep van Python functie `choice()` in de `numpy.random` klasse. De functie is gebaseerd op een algoritme van D.E. Knuth, zie p. 142 van zijn boek *The Art of Computer Programming, Volume 2 (Seminumerical Algorithms)*, Third Edition (1997). De basisvorm van dit algoritme staat in paragraaf 5.1.3 van het boek.