Pseudo Random Nummer Generator

Ik heb mijn generator 1000 willekeurige getallen laten genereren tussen de 0 en de 1 (met verschillende seeds, van 1 tot 100). Hiervan heb ik daarna het gemiddelde genomen. Als kleinste gemiddelde kwam hier uit: 0.4868943943943945. Als grootste gemiddelde kwam hier uit: 0.5107682682682692.

Natuurlijk kan je met alleen een gemiddelde niet veel zeggen over de distributie. Immers kan het zo zijn dat alle getallen bijv. tussen de 0.4 en 0.6 liggen, en dan heb je geen uniforme verdeling van 0 tot 1.

Dus moest ik naar de uitkomst van de functie ‘chisquare’ kijken. Eerst heb ik een numpy array aangemaakt. Hierin heb ik intervallen ingezet die ieder 0.1 groot zijn, dus dat zijn in totaal 100 intervallen (van 0 tot 1).

Vervolgens heb ik alle gegenereerde getallen in het juiste interval geplaatst. Daarna heb ik geteld hoeveel getallen er voor kwamen in ieder interval. Het aantal getallen per interval heb ik daarna geplaatst in een array, en deze heb ik daarna als argument aan de chisquare functie meegegeven.

H0: de generator is uniform tussen 0 en 1 verdeeld

H1: de generator is niet uniform tussen 0 en 1 verdeeld

Aantal getallen: 1000

Seed: 100

Statistic: 1.66

P-value: 0.9957766436653857

Er is dus 1.66% kans dat de gegenereerde getallen door kans uniform verdeeld zijn. De p-waarde is 0.99577 etc. Dit is dus veel groter dan 0.05. De null hypothese wordt niet verworpen en we mogen er dus van uit gaan dat onze generator voor uniform verdeelde getallen tussen de 0 en 1 zorgt.

Om hier toch nog iets zeker van te zijn testen we een andere seed:

Aantal getallen: 1000

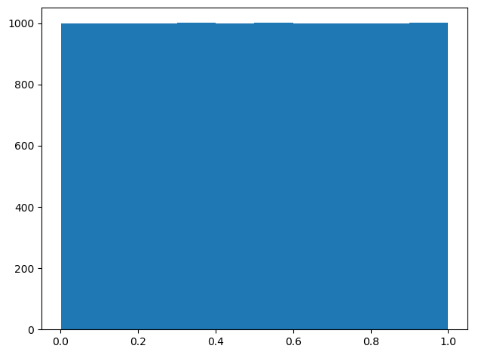
Seed: 5

Statistic: 2.1999999999999997

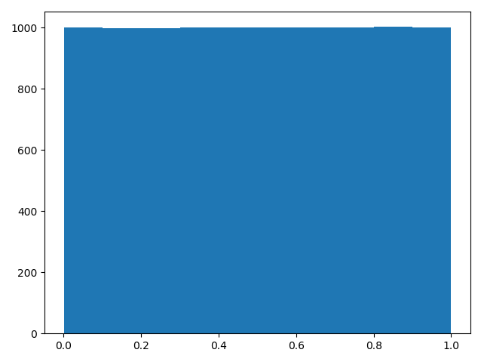
P-value: 0.9878955401745281

Er is 2.1999% kans dat de gegenereerde getallen door kans uniform verdeeld zijn. De p-waarde is 0.987, wat dus opnieuw veel groter is dan 0.05. De null hypothese wordt dus niet verworpen.

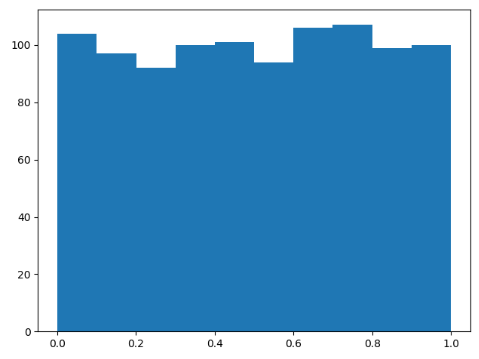
Opmerking: de seed mag niet te hoog zijn. Hoe dichter de seed de variabele ‘m’ nadert, hoe slechter de generator zich af en toe zal gedragen. Dit was vooral van belang bij het benaderen van Pi. De seed mocht niet te hoog zijn, anders kon de generator niet Pi benaderen!



Seed = 10, Aantal getallen = 10,000



Seed = 600, Aantal getallen = 10,000



Seed = 600, Aantal getallen = 1000

Hoe meer getallen je hebt, des te dichter in de buurt kom je van een echte uniforme verdeling.