



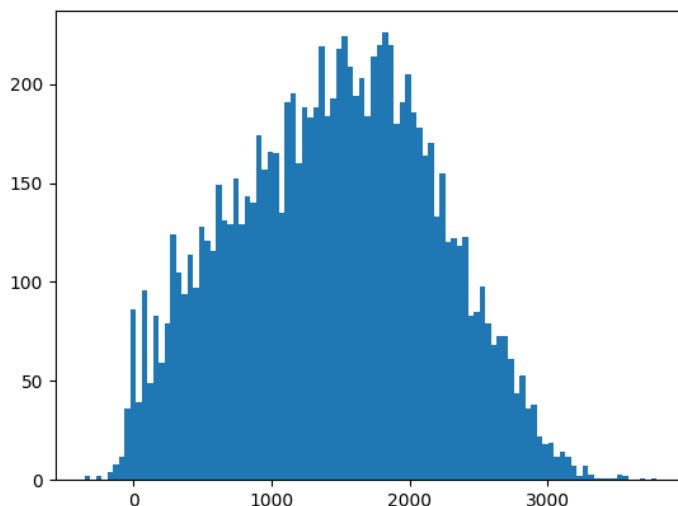
Baseline

Door: Team-trein

De random traject keuze is gebaseerd op 4 random generators in python, dit zijn ook onze plekken waar bias voorkomen. Allereerst kiezen wij random een getal tussen 1 en 7 (1 en 20 voor heel Nederland) om het aantal gebruikte trajecten te bepalen. Dit levert een grote bias op aangezien wij net zoveel kans geven voor het gebruik van een traject als 7 terwijl er in de state-space veel meer opties met 7 dan met 1 zitten. Hierna genereren wij een random nummer om het start station te bepalen. Dit levert ook een bias op aangezien alle stations dezelfde kans hebben, terwijl sommige veel meer mogelijkheden hebben voor een uniek traject, waardoor ze dus vaker voorkomen in de state space.

Verder kiezen wij vanaf elk station een random keuze om een bepaalde kant op te gaan. Dit levert ook een kleine bias op aangezien de korte stukken vaker voorkomen in de state space dan de lange stukken, aangezien deze minder vervolg connecties kunnen hebben vanwege de tijdsbeperking, die ervoor zorgt dat je minder mogelijkheden hebt.

Verder hebben we een op 10 kans om te stoppen met het traject op elk punt. Dit zorgt ervoor dat er bijvoorbeeld heel veel trajecten zitten met maar een stop in onze random output. Veel meer dan er daadwerkelijk in de statespace zitten. Aangezien er vrij weinig combinaties zijn met maar een stop ten opzichte van hoe vaak wij het tegenkomen.



Y as staat hoe vaak een score voorkwam, x-as de scores zelf, grafiek is met 100 bins gegenereerd.

Dan de daadwerkelijke verdeling van onze random gegenereerde oplossingen. Hiervoor hebben wij 10000 iteraties gedaan op de dataset van heel Nederland. Hierbij kregen wij een gemiddelde van 1450,7 als waarde voor k. Wij hebben verder een histogram gemaakt met de resultaten. Hier zien wij dat de meeste resultaten onder het gemiddelde

zitten, maar dat er enkele uitschieters zijn naar boven. Dit is relevant aangezien het betekent dat de random generator kan worden gebruikt om oplossingen te genereren die daadwerkelijk bruikbaar zijn als we het lang genoeg draaien. Dit hebben wij ook gemerkt bij het draaien op de kleinere dataset, aangezien wij daar toen wij het een miljoen keer draaide al een resultaat vonden met een p waarde van 1.

Verder hebben wij in python getest of het een bekende distributie volgt, maar wij hebben geen resultaten gevonden die dat bevestigen.